

OpenEye
SCIENTIFIC

OEDepict TK – Java

Release 2026.1.0

Cadence Design Systems, Inc.

August 06, 2026

CONTENTS

1 Introduction **1**

2 Theory **3**
2.1 Basic Drawing Primitives . . . 3
2.2 Molecule Depiction . . . 6
2.3 Highlighting . . . 24
2.4 Molecule Layouts . . . 31
2.5 Multi Page Images . . . 39
2.6 MDL Query Depiction . . . 42
2.7 MDL Reaction Depiction . . . 49
2.8 Molecule Alignment . . . 51
2.9 Generating Interactive SVG Images . . . 62

3 Examples **69**
3.1 OEDepict Examples . . . 69
3.2 Python Cookbook Examples . . . 119

4 API **121**
4.1 OEDepict Classes . . . 121
4.2 OEDepict Constants . . . 414
4.3 OEDepict Functions . . . 547

5 Appendices **627**
5.1 Appendix: Element coloring (CPK) . . . 627

6 Release History **631**
6.1 OEDepict TK 2.7.0 . . . 631
6.2 OEDepict TK 2.6.0 . . . 631
6.3 OEDepict TK 2.5.6 . . . 631
6.4 OEDepict TK 2.5.5 . . . 631
6.5 OEDepict TK 2.5.4 . . . 631
6.6 OEDepict TK 2.5.3 . . . 631
6.7 OEDepict TK 2.5.2 . . . 631
6.8 OEDepict TK 2.5.1 . . . 631
6.9 OEDepict TK 2.5.0 . . . 632
6.10 OEDepict TK 2.4.7 . . . 632
6.11 OEDepict TK 2.4.6 . . . 632
6.12 OEDepict TK 2.4.5 . . . 632
6.13 OEDepict TK 2.4.4 . . . 632
6.14 OEDepict TK 2.4.3 . . . 632
6.15 OEDepict TK 2.4.2 . . . 633

6.16 OEDepict TK 2.4.1 . . . 633
6.17 OEDepict TK 2.4.0 . . . 635
6.18 OEDepict TK 2.3.6 . . . 635
6.19 OEDepict TK 2.3.5 . . . 638
6.20 OEDepict TK 2.3.4 . . . 639
6.21 OEDepict TK 2.3.3 . . . 641
6.22 OEDepict TK 2.3.2 . . . 641
6.23 OEDepict TK 2.3.1 . . . 642
6.24 OEDepict TK 2.3.0 . . . 643
6.25 OEDepict TK 2.2.6 . . . 647
6.26 OEDepict TK 2.2.5 . . . 648
6.27 OEDepict TK 2.2.4 . . . 651
6.28 OEDepict TK 2.2.3 . . . 651
6.29 OEDepict TK 2.2.2 . . . 653
6.30 OEDepict TK 2.2.1 . . . 654
6.31 OEDepict TK 2.2.0 . . . 655
6.32 OEDepict TK 2.1.0 . . . 656
6.33 OEDepict TK 2.0.4 . . . 658
6.34 OEDepict TK 2.0.3 . . . 660
6.35 OEDepict TK 2.0.2 . . . 662
6.36 OEDepict TK 2.0.1 . . . 663
6.37 OEDepict TK 2.0.0 . . . 663

7 Copyright and Trademarks **667**

8 Sample Code **669**

9 Citation **671**
9.1 Orion® . . . 671
9.2 Orion Floes . . . 671
9.3 Toolkits and Applications . . . 671
9.4 Publications for Bibliographies . . . 673
9.5 OpenEye MMDS Web Service . . . 674

10 Technology Licensing **675**
10.1 GCC . . . 747

Index **763**

CHAPTER

ONE

INTRODUCTION

OEDepict TK provides functionalities to render chemical structures and output molecule depictions to a wide range of graphical file formats.

In addition to standard depiction routines, **OEDepict TK** provides a powerful graphics engine which supports:

- *Basic shape drawing*
- *Highly customizable molecule depiction*
- *Variety of highlighting styles*
- *Customizable layout options*
- *Depictions aligned by MCS or substructure*
- *MDL Query* and *MDL Reaction* depictions

CHAPTER

TWO

THEORY

2.1 Basic Drawing Primitives

The following example introduces how an image can be created in **OEDepict TK** to generate a single black line across an image:

1. Create an *OEImage* object with a specific width and height.
2. Specify the two end points of the line with *OE2DPoint* objects.
3. Draw the line with the `OEImage.DrawLine` method using the `OEDepict.OEBlackPen` to specify the graphical properties of the line.
4. Write the image into a file by calling the `OEWriteImage` function.

   The *OEImage* class is implemented as a container of drawing commands. When a method such as `OEImage.DrawLine` is called, rather than immediately drawing the line specified by the parameters, a line drawing operation is created in the *OEImage* class for later execution.

Listing 1: Example of line drawing

```
public class DrawLine {
    public static void main(String argv[]) {
        int width  = 100;
        int height = 100;
        // Create image
        OEImage image = new OEImage(width, height);

        // Draw line with default pen
        image.DrawLine(new OE2DPoint(10.0, 10.0),
                new OE2DPoint(90.0, 90.0),
                oedepict.getOEBlackPen());

        // Write image to SVG file
        oedepict.OEWriteImage("DrawLine.svg", image);
    }
}
```

The image created by `Listing 1` is shown in *Figure: Example of line drawing*.

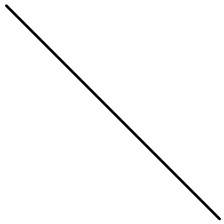


Fig. 1: **Example of line drawing**

See also

- *OEPen* class
- *OEImage* class
- `OEWriteImage` function

2.1.1 Coordinate System

The coordinate system used in **OEDepict TK** has the origin (x=0.0, y=0.0) at the top left with the x-axis pointing to the right and the y-axis pointing down.

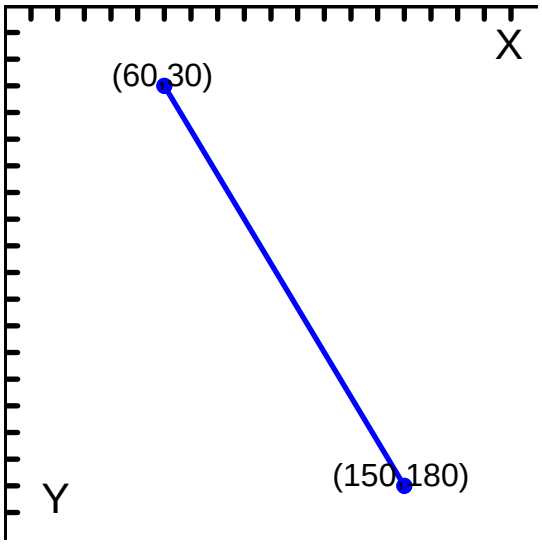


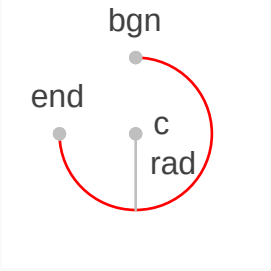
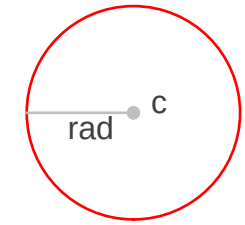
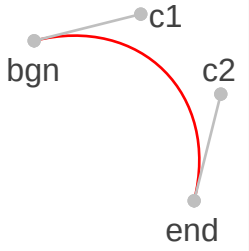
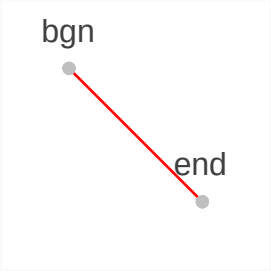
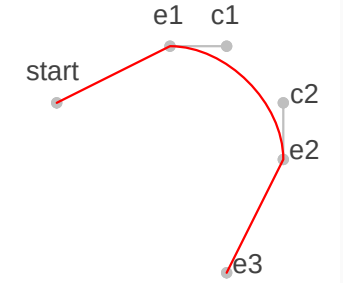
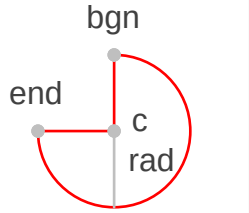
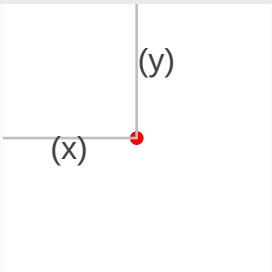
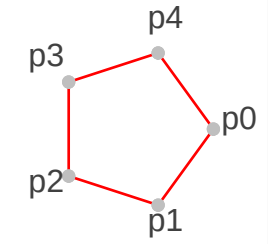
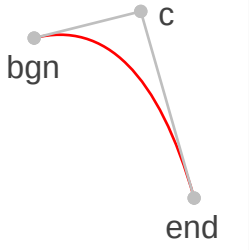
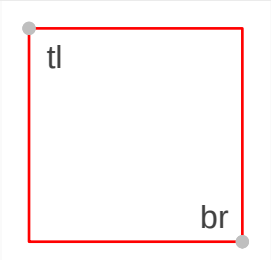
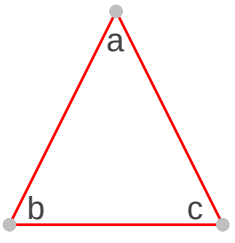
Fig. 2: **2D coordinate system** *(X = horizontal, Y = vertical)*

See also

- *OE2DPoint* class

2.1.2 Basic Shapes

OEDepict TK supports drawing the following basic shapes:

OEImageBase.DrawArc	*OEImageBase.DrawCircle*	*OEImageBase.DrawCubicBezier*
		
OEImageBase.DrawLine	*OEImageBase.DrawPath*	*OEImageBase.DrawPie*
		
OEImageBase.DrawPoint	*OEImageBase.DrawPolygon*	*OEImageBase.DrawQuadraticBezier*
		
OEImageBase.DrawRectangle	*OEImageBase.DrawText*	*OEImageBase.DrawTriangle*
		

2.1.3 Image File Formats

OEDepict TK natively supports a variety of graphical file formats. `OEWriteImage` automatically detects the desired file format from the file name passed to the function. The following table lists the file formats natively supported in **OEDepict TK** and their associated file extensions:

Graphics File Format	Format Type	File Extension
PNG (Portable Network Graphics)	raster image	`.png`
SVG (Scalable Vector Graphics)	vector image	`.svg`
bare SVG (with no header)	vector image	`.bsvg`
Postscript	vector image	`.ps`
Encapsulated PostScript	vector image	`.eps`
PDF (Portable Document Format)	vector image	`.pdf`

The `OEIsRegisteredImageFile` function can be used to query whether a given file extension is supported by **OEDepict TK**. All the supported file formats can be queried using the `OEGetSupportedImageFileExtensions` function.

2.2 Molecule Depiction

The previous chapter introduced how to create images and draw basic geometric elements (such as text, lines rectangles, etc.) Drawing these basic graphical elements provides a framework to construct more complex images such as molecular diagrams.

The molecule depiction is divided into two separate steps (see *Figure: Process of Molecule Depiction*):

1. Generating 2D coordinates
2. Rendering molecule that involves
    - transforming and scaling coordinates in order to fit the molecule into an image with a specific width and height
    - setting atom and bond display properties based on depiction style
    - creating an image (such as drawing lines for bonds, drawing text for atomic labels)

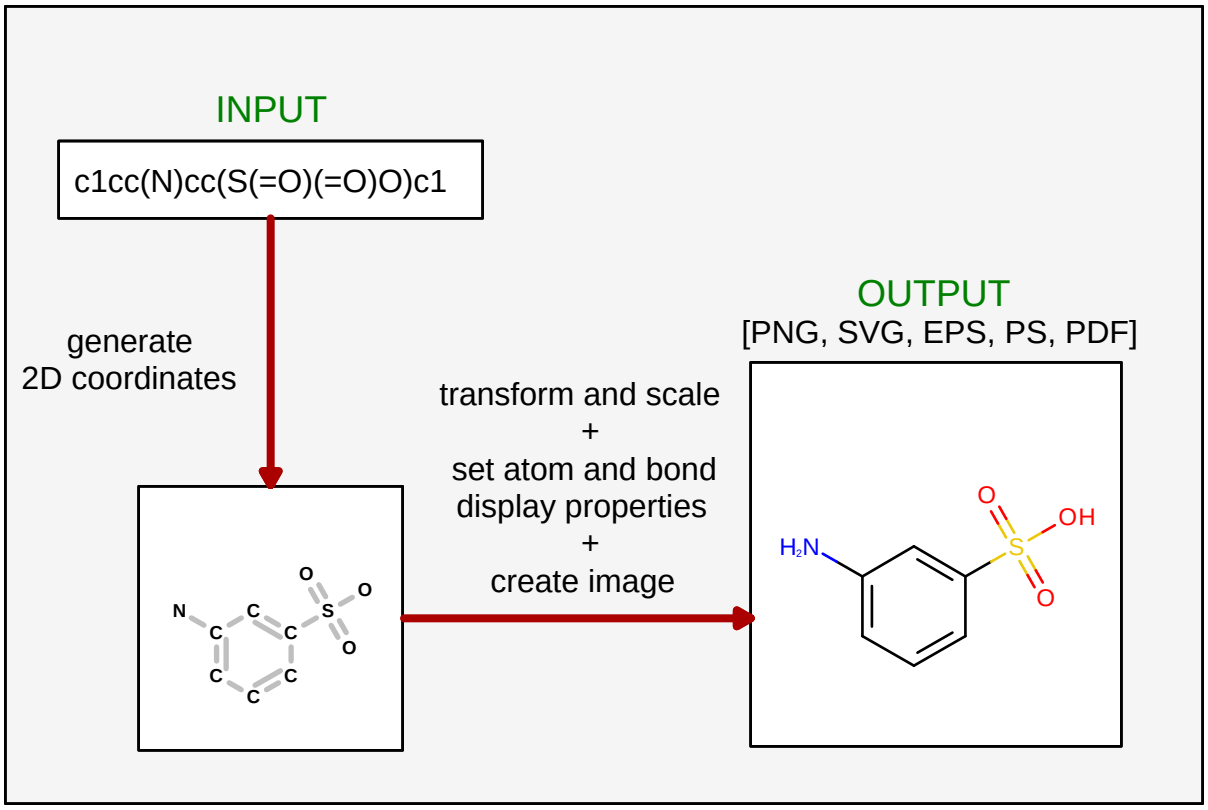


Fig. 3: **Process of molecule depiction**

The next example (*Listing 1*) demonstrates how easy is to depict a molecule using the **OEDepict TK**. After creating a molecule (for example by parsing a SMILES string) the image can be generated by calling two functions.

1. The *OEPrepareDepiction* function prepares the molecule for depiction. This process may involve perceiving atom and bond stereo information and suppressing or adding hydrogens. It also calls the *OEDepictCoordinates* function to generate the 2D coordinates of the molecule.
2. The *OERenderMolecule* function generates the image of the molecule using the default depiction styles and then writes the image into the given file.

The image created by *Listing 1* is shown in *Figure: Example of depicting a molecule*.

Listing 1: Example of molecule depiction

```
public class DepictMolSimple {
    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cc(N)cc(S(=O)(=O)O)c1 3-aminobenzenesulfonic acid");
        oedepict.OEPrepareDepiction(mol);
        oedepict.OERenderMolecule("DepictMolSimple.svg", mol);
    }
}
```

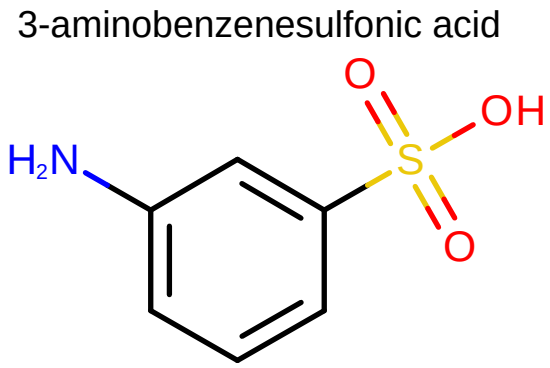


Fig. 4: **Example of molecule depiction**

See also

- *OEPrepareDepiction* function
- *OERenderMolecule* function

2.2.1 Customizing Molecule Depiction

The previous example showed how to generate a molecule diagram with default parameters. **OEDepict TK** provides the ability to customize molecule depiction. The *Listing 2* example shows how to depict a molecule width specific width and height parameters.

Listing 2: Example of depicting a molecule

```
public class DepictMolSize {
    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cc(N)cc(S(=O)(=O)O)c1 3-aminobenzenesulfonic acid");
        oedepict.OEPrepareDepiction(mol);

        double width  = 300;
        double height = 300;
        double scale  = OEScale.AutoScale;
        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
        oedepict.OERenderMolecule("DepictMolSize.svg", disp);
    }
}
```

See also

- *OE2DMolDisplayOptions* class
- *OE2DMolDisplay* class

Apart from defining the dimensions of a depicted molecule, the *OE2DMolDisplayOptions* class stores all properties that determine how a molecule is depicted. The following example shows how to set various depiction styles before

initializing the *OE2DMolDisplay* object. The image created by `Listing 3` is shown in *Figure: Example of customizing depiction style*.

Listing 3: Example of customizing the depiction style

```
public class DepictMolStyle {
    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cc(N)cc(S(=O)(=O)O)c1 3-aminobenzenesulfonic acid");
        oedepict.OEPrepareDepiction(mol);

        double width  = 200;
        double height = 200;
        double scale  = OEScale.AutoScale;
        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);

        opts.SetAromaticStyle(OEAromaticStyle.Circle);
        opts.SetAtomColorStyle(OEAtomColorStyle.BlackCPK);
        opts.SetTitleLocation(OETitleLocation.Bottom);

        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
        oedepict.OERenderMolecule("DepictMolStyle.svg", disp);
    }
}
```

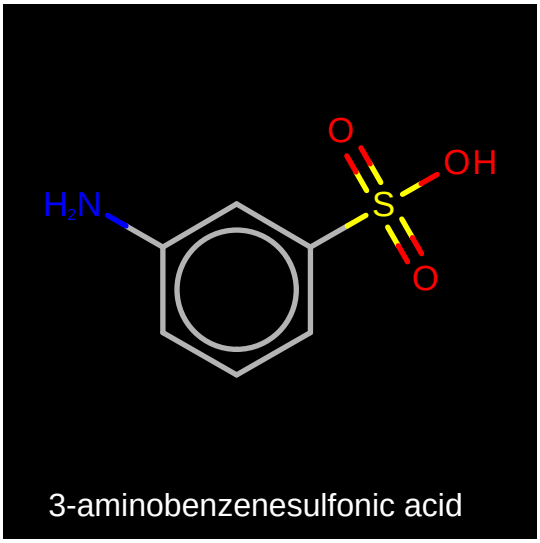


Fig. 5: **Example of customizing the depiction style**

Apart from controlling the depiction style through the *OE2DMolDisplayOptions* class, **OEDepict TK** also provides access to manipulate individual atom and bond 2D display properties. The `OE2DMolDisplay.GetAtomDisplays` method returns an iterator over *OE2DAtomDisplay* objects that store atom display information. Similarly, the `OE2DMolDisplay.GetBondDisplays` method returns an iterator over *OE2DBondDisplay* objects that store bond display information.

The following example demonstrates how to change atom and bond display properties and depict the graph of the molecule by ignoring atom labels and setting bond display styles. The image created by `Listing 4` is shown in *Figure: Example of molecule graph depiction*.

Listing 4: Example of molecule graph depiction

```
public class DepictMolGraph {
    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cc(N)cc(S(=O)(=O)O)c1");
        oedepict.OEPrepareDepiction(mol);

        double width  = 200;
        double height = 200;
        double scale  = OEScale.AutoScale;
        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);

        opts.SetAtomColorStyle(OEAtomColorStyle.WhiteMonochrome);
        opts.SetTitleLocation(OETitleLocation.Hidden);
        opts.SetAtomColor(OEElemNo.C, oechem.getOEDarkGreen());
        OEPen pen = new OEPen(oechem.getOEBlack(), oechem.getOEBlack(), OEFill.On, 3.0);
        opts.SetDefaultBondPen(pen);

        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);

        for (OE2DAtomDisplay adisp : disp.GetAtomDisplays()) {
            adisp.SetLabel("");
        }

        for (OE2DBondDisplay bdisp : disp.GetBondDisplays()) {
            bdisp.SetDisplayType(OEBondDisplayType.Single);
        }

        oedepict.OERenderMolecule("DepictMolGraph.svg", disp);
    }
}
```

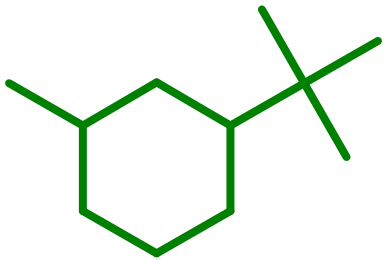


Fig. 6: **Example of molecule graph depiction**

Hint

Even though **OEDepict TK** provides the ability to manipulate atom and bond display properties of after the *OE2DMolDisplay* object is constructed, it is highly recommended to set the depiction properties using the *OE2DMolDisplayOptions* class. Only by knowing all properties (such as labels, font styles and sizes *etc.*) in advance can one ensure the **best depiction layout** *i.e.* that the molecule diagram is rendered without any label clippings and the labels are displayed with the minimum number of overlaps.

See also

- *OE2DAtomDisplay* class
- *OE2DBondDisplay* class

2.2.2 Controlling the Size of Depicted Molecules

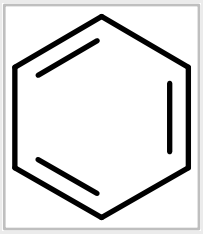
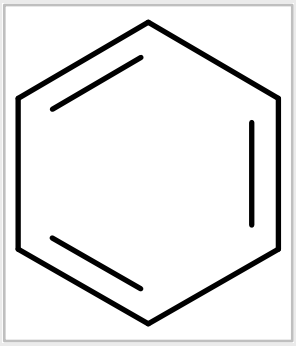
The dimensions of a depicted molecule are controlled by the width, height and scale parameters of the *OE2DMolDisplayOptions* that is used to initialized the *OE2DMolDisplay* object. (See example in `Listing 2`)

See also

- `OE2DMolDisplayOptions.SetDimensions` method
- `OE2DMolDisplayOptions.SetHeight` method
- `OE2DMolDisplayOptions.SetScale` method
- `OE2DMolDisplayOptions.SetWidth` method

If the width and the height of the image is not specified (set to be zero), then scaling value determine the dimensions of the image. (See examples in *Examples of image scaling*). The default scale is defined to be `OEScale.Default` constant, and the image may be enlarged or shrunk by specifying a floating point scaling value. For example, the scaling `OEScale.Default` * `0.5` generates an image a half of the default size, and scaling `OEScale.Default` * `1.5` generates an image one and a half times the default.

Table 1: **Examples of image scaling** *(width = 0.0, height = 0.0)*

OEScale::Default * 0.5	OEScale::Default	OEScale::Default * 1.5
		

If an *OE2DMolDisplayOptions* object is initialized with zero width and / or height, then the real dimensions of the image can be accessed after creating an *OE2DMolDisplay* object.

```
double width  = 0.0;
double height = 0.0;
double scale  = 50.0;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
System.out.format("width  %.1f\n", disp.GetWidth());
System.out.format("height %.1f\n", disp.GetHeight());
System.out.format("scale  %.1f\n", disp.GetScale());
```

The output of the preceding snippet is the following:

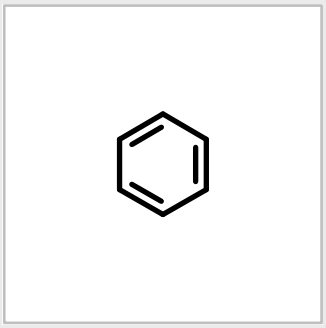
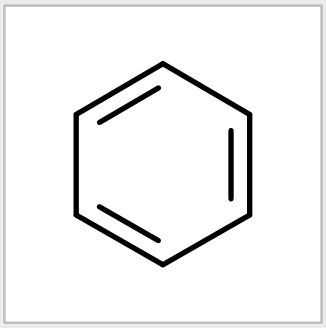
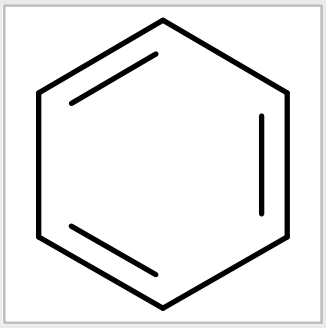
```
width  137.0
height 181.0
scale  50.0
```

See also

OE2DMolDisplay.GetHeight method *OE2DMolDisplay.GetWidth* method *OE2DMolDisplay.GetScale* method

If both the width and the height of are specified, then *OEScale.AutoScale* scaling indicates that the molecule is scaled to maximally fit the given dimensions. The specified width and height will not be altered, not even if the given scaling would require it *i.e* if necessary the scaling will be reduced in order to fit the molecule into the given dimension. (See examples in *Examples of image scaling*).

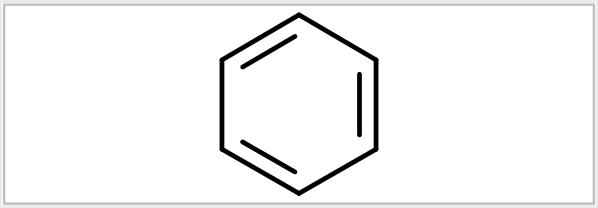
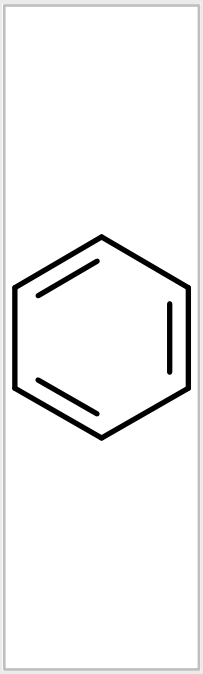
Table 2: **Examples of fixed sized image** *(width = 120.0, height = 120.0)*

OEScale::Default * 0.5	OEScale::Default	OEScale::Default * 1.5
		

Note

Specifying a width and / or height larger than what is required by the scale, causes the molecule to float in the middle of the image of the requested size.

If only either the width or the height of the image is specified, then the unspecified dimension will be set based on the scaling and the specified width / height, respectively.

OEScale::Default, width = 250.0, height = 0.0	OEScale::Default, width = 0.0, height = 250.0
	

2.2.3 Molecule Depiction with Transparent Background

The following code snippet shows how to generate a molecule depiction with transparent background. First, the *OEImage* object, onto which the depiction is going to be rendered, has to be constructed with the `OETransparentColor` color. Then the *OERenderMolecule* function has to be called with a parameter that ensures that the background in not cleared prior to rendering the molecule.

```
double width  = 200;
double height = 200;
double scale  = OEScale.AutoScale;
OEImage image = new OEImage(width, height, oechem.getOETransparentColor());

OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);

boolean clearbackground = true;
oedepict.OERenderMolecule("DepictMolTransparent.png", disp, !clearbackground);
```

2.2.4 Displaying Atom Properties

OEDepict TK provides ability the display atom properties as strings. When a *OE2DMolDisplay* object is created, the positions where atom properties can be displayed are calculated. (see *Figure: Example of generating atom property positions*.)

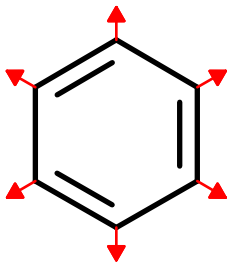


Fig. 7: **Example of generating atom property positions**

The following example shows how to display the atom indices by using the *OEDisplayAtomIdx* functor. The font used to depict the atom property strings also can be modified by using the *OE2DMolDisplayOptions.SetAtomPropLabelFont* method. The image created by *Listing 5* is shown in *Figure: Example of displaying atom indices*.

Listing 5: Example of depicting atom indices

```
public class AtomPropIndex {
    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cc(N)cc(S(=O)(=O)O)c1");
        oedepict.OEPrepareDepiction(mol);

        double width  = 300;
        double height = 200;
        double scale  = OEScale.AutoScale;

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);

        opts.SetAtomPropertyFunctor(new OEDisplayAtomIdx());
        opts.SetAtomPropLabelFont(new OEFont(oechem.getOEDarkGreen()));

        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
        oedepict.OERenderMolecule("AtomPropIndex.svg", disp);
    }
}
```

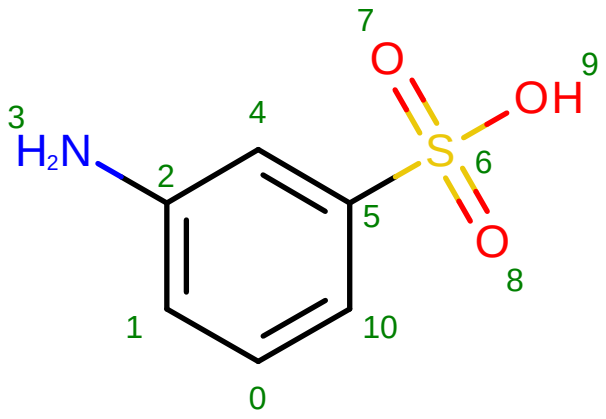


Fig. 8: **Example of displaying atom indices**

Hint

It is recommended to use the *OE2DMolDisplayOptions.SetAtomPropertyFunctor* method to define the atom property labels when the *OE2DMolDisplay* object is constructed. All labels displayed on the molecule diagram have to be known in advance in order to be able to minimize the number of label clashes and clippings when calculating the positions of the atom property labels.

See also

OEDepict TK provides the following built-in atom property functors:

- *OEDisplayAtomIdx* class
- *OEDisplayAtomMapIdx* class
- *OEDisplayNoAtomProp* class

In order to display user-defined atom properties, the functor that is passed to the *OE2DMolDisplayOptions.SetAtomPropertyFunctor* method has to be derived from the *OEDisplayAtomPropBase* abstract base class. The *Listing 6* example shows how to implement a user-defined functor that marks aromatic atoms. The image created by *Listing 6* is shown in *Figure: Example of displaying user-defined atom properties*.

Listing 6: Example of depicting user-defined atom properties

```
public class AtomPropUser {
    private static class LabelAromaticAtoms extends OEDisplayAtomPropBase {
        public LabelAromaticAtoms() {}

        @Override
        public String constCall(OEAtomBase atom) {
            return (atom.IsAromatic() ? "(A)" : "");
```

(continues on next page)

(continued from previous page)

```
        }

        @Override
        public OEDisplayAtomPropBase CreateCopy() {
            OEDisplayAtomPropBase copy = new LabelAromaticAtoms();
            copy.swigReleaseOwnership();
            return copy;
        }
    }

    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cc(N)cc(S(=O)(=O)O)c1");
        oedepict.OEPrepareDepiction(mol);

        double width  = 300;
        double height = 200;
        double scale  = OEScale.AutoScale;

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);

        LabelAromaticAtoms atomlabel = new LabelAromaticAtoms();
        opts.SetAtomPropertyFunctor(atomlabel);

        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
        oedepict.OERenderMolecule("AtomPropUser.svg", disp);
    }
}
```

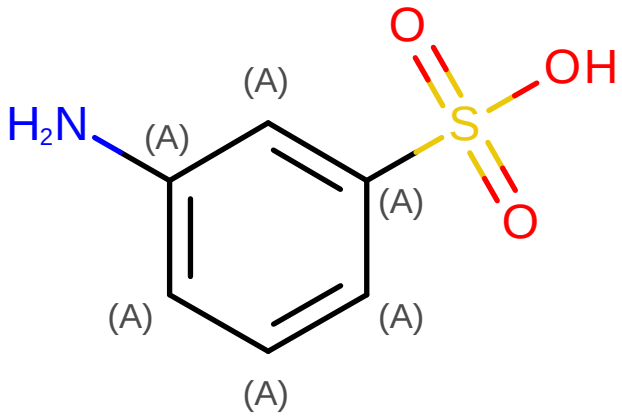


Fig. 9: **Example of displaying user-defined atom properties**

See also

- *OEDisplayAtomPropBase* class

2.2.5 Displaying Bond Properties

Similarly to atom properties, bond properties can be displayed as strings. (see *Figure: Example of generating bond property positions*.)

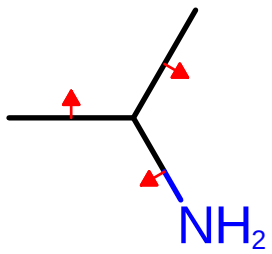


Fig. 10: **Example of generating bond property positions**

The *Listing 7* demonstrates how to display bond indices by using the *OEDisplayBondIdx* functor. The font used to depict the bond property strings can be set by using the *OE2DMolDisplayOptions.SetBondPropLabelFont* method. The image created by *Listing 7* is shown in *Figure: Example of displaying bond indices*.

Listing 7: Example of depicting bond indices

```
public class BondPropIndex {
    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cc(N)cc(S(=O)(=O)O)c1");
        oedepict.OEPrepareDepiction(mol);

        double width  = 300;
        double height = 200;
        double scale  = OEScale.AutoScale;

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);

        opts.SetBondPropertyFunctor(new OEDisplayBondIdx());
        opts.SetBondPropLabelFont(new OEFont(oechem.getOEDarkBlue()));

        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
        oedepict.OERenderMolecule("BondPropIndex.svg", disp);
    }
}
```

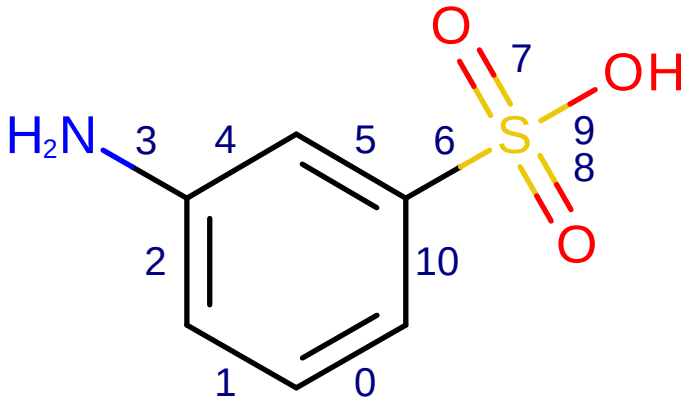


Fig. 11: **Example of displaying bond indices**

Hint

It is recommended to use the *OE2DMolDisplayOptions.SetBondPropertyFunctor* method to define the bond property labels when the *OE2DMolDisplay* object is constructed. All labels displayed on the molecule diagram have to be known in advance in order to be able to minimize the number of label clashes and clippings when calculating the positions of the bond property labels.

See also

OEDepict TK provides the following built-in bond property functors:

- *OEDisplayBondIdx* class
- *OEDisplayNoBondProp* class

In order to display user-defined bond properties, the functor that is passed to the *OE2DMolDisplayOptions.SetBondPropertyFunctor* method has to be derived from the *OEDisplayBondPropBase* abstract base class. The *Listing 8* example shows how to implement a user-defined functor that marks chain and ring bonds. The image created by *Listing 8* is shown in *Figure: Example of displaying user-defined bond properties*.

Listing 8: Example of depicting user-defined bond properties

```
public class BondPropUser {
    private static class LabelRingChainBonds extends OEDisplayBondPropBase {
        public LabelRingChainBonds() {}

        @Override
        public String constCall(OEBondBase bond) {
            return (bond.IsInRing() ? "R" : "C");
        }
```

(continues on next page)

(continued from previous page)

```
        @Override
        public OEDisplayBondPropBase CreateCopy() {
            OEDisplayBondPropBase copy = new LabelRingChainBonds();
            copy.swigReleaseOwnership();
            return copy;
        }
    }

    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cc(N)cc(S(=O)(=O)O)c1");
        oedepict.OEPrepareDepiction(mol);

        double width  = 300;
        double height = 200;
        double scale  = OEScale.AutoScale;

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);

        LabelRingChainBonds bondlabel = new LabelRingChainBonds();
        opts.SetBondPropertyFunctor(bondlabel);

        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
        oedepict.OERenderMolecule("BondPropUser.svg", disp);
    }
}
```

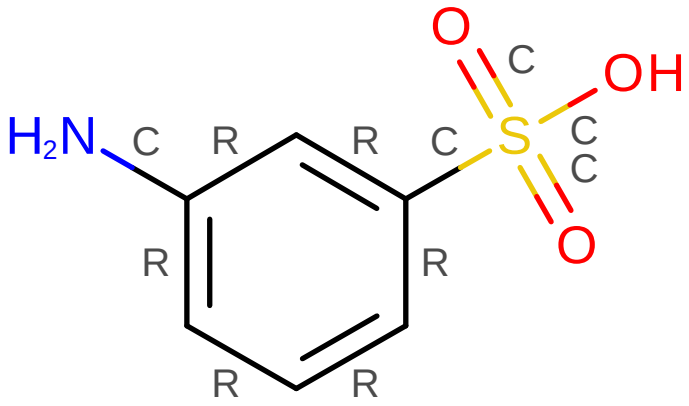


Fig. 12: **Example of displaying user-defined bond properties**

See also

- *OEDisplayBondPropBase* class

2.2.6 Hiding Atoms and Bonds

OEDepict TK provides ability the suppress the display of specific atom and bonds. When a *OE2DMolDisplay* object is created, the visibility of every atom is evaluated. Bonds with both atom endpoints visible are also displayed. (see *Figure: Example of hiding a portion of a structure*.)

The following example shows how to selectively display only a portion of the structure by using a custom class derived from the `OEUnaryAtomPred` functor. The images created by *Listing 9* is shown in *Figure: Example of hiding a portion of a structure*.

Hint

It is recommended to use the *OE2DMolDisplayOptions.SetAtomVisibilityFunctor* method to identify the visible atoms when the *OE2DMolDisplay* object is constructed. All objects displayed on the molecule diagram have to be known in advance in order to be able to correctly size and scale the molecule.

Listing 9: Example of selectively displaying atoms

```
public class AtomVisibility {

    static class OEAtomVisibilityShowRxnRole extends OEUnaryAtomPred {
        private int rxnrole = OERxnRole.None;

        public OEAtomVisibilityShowRxnRole(int role) { rxnrole = role; }

        @Override
        public boolean constCall(OEAtomBase atom) {
            if (rxnrole == OERxnRole.None) {
                    return true;
            }
            return (rxnrole == atom.GetRxnRole());
        }

        @Override
        public OEUnaryAtomPred CreateCopy() {
            OEUnaryAtomPred copy = new OEAtomVisibilityShowRxnRole(rxnrole);
            copy.swigReleaseOwnership();
            return copy;
        }
    }

    public static void main(String argv[]) {
        if (argv.length != 2) {
            oechem.OEThrow.Usage("AtomVisibility <rxnfile> <imagefile>");
        }
        oemolistream ifs = new oemolistream(argv[0]);
```

(continues on next page)

(continued from previous page)

```
        OEGraphMol mol = new OEGraphMol();
        if (!oechem.OEReadRxnFile(ifs,mol)) {
            oechem.OEThrow.Fatal("error reading rxnfile");
        }
        ifs.close();

        oedepict.OEPrepareDepiction(mol);

        OEImage image = new OEImage(900, 100);

        int rows = 1;
        int cols = 3;
        OEImageGrid grid = new OEImageGrid(image, rows, cols);

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(grid.GetCellWidth(), grid.
↪GetCellHeight(), OEScale.AutoScale);

        OEImageBase cell = grid.GetCell(1,1);
        mol.SetTitle("Reaction display");
        OEUnaryAtomPred allatoms = new OEIsTrueAtom(); // explicitly set the default
        opts.SetAtomVisibilityFunctor(allatoms);
        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
        double rxnscale = disp.GetScale();
        oedepict.OERenderMolecule(cell, disp);

        cell = grid.GetCell(1,2);
        mol.SetTitle("Reactant display");
        OEAtomVisibilityShowRxnRole showreacs = new␣
↪OEAtomVisibilityShowRxnRole(OERxnRole.Reactant);
        opts.SetAtomVisibilityFunctor(showreacs);
        opts.SetScale(rxnscale);
        disp = new OE2DMolDisplay(mol, opts);
        oedepict.OERenderMolecule(cell, disp);

        cell = grid.GetCell(1,3);
        mol.SetTitle("Product display");
        OEAtomVisibilityShowRxnRole showprods = new␣
↪OEAtomVisibilityShowRxnRole(OERxnRole.Product);
        opts.SetAtomVisibilityFunctor(showprods);
        opts.SetScale(rxnscale);
        disp = new OE2DMolDisplay(mol, opts);
        oedepict.OERenderMolecule(cell, disp);

        oedepict.OEWriteImage(argv[1], image);
    }
}
```

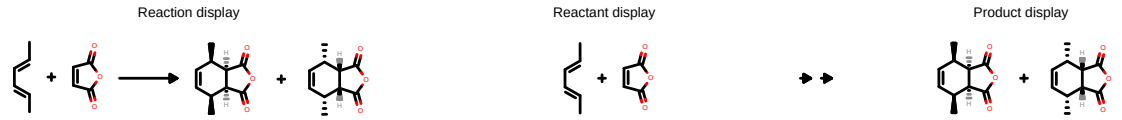


Fig. 13: **Example of hiding a portion of a structure**

2.2.7 Configuring Molecule Display

OEDepict TK provides a mechanism to generate a standard interface that allows configuration of molecule displays. By calling the *OEConfigureImageOptions* function, "-height" and "width" parameters are added to the interface (OEInterface), at run-time the value of these parameters can be obtained by calling the *OEGetImageHeight* and *OEGetImageWidth* functions, respectively. (See the standard interface displayed when calling the *Listing 10* example with the "–help" parameter)

The *OEConfigure2DMolDisplayOptions* function generates an interface that allows setting of various molecule display options (such as aromatic style, atom and bond coloring etc.) The *OE2DMolDisplayOptions* object, that stores properties determine the molecule depiction style, can initialized based on to the command line parameters by calling the *OESetup2DMolDisplayOptions* function.

Listing 10: Example of configuring mol display

```
public class DepictMolConfigure {
    static String InterfaceData =
            "!CATEGORY \"input/output options :\"\n" +
                    "\n" +
                    "  !PARAMETER -in\n" +
                    "    !ALIAS -i\n" +
                    "    !TYPE string\n" +
                    "    !REQUIRED true\n" +
                    "    !BRIEF Input filename\n" +
                    "  !END\n" +
                    "\n" +
                    "  !PARAMETER -out\n" +
                    "    !ALIAS -o\n" +
                    "    !TYPE string\n" +
                    "    !REQUIRED true\n" +
                    "    !BRIEF Output filename\n" +
                    "  !END\n" +
                    "\n" +
                    "!END\n";

    public static void main(String argv[]) {
        OEInterface itf = new OEInterface();
        // import configuration file
        oechem.OEConfigure(itf, InterfaceData);
        // add configuration for image size and display options
        oedepict.OEConfigureImageOptions(itf);
        oedepict.OEConfigure2DMolDisplayOptions(itf);

        if (!oechem.OEParseCommandLine(itf, argv, "DepictMolConfigure")) {
            oechem.OEThrow.Fatal("Unable to interpret command line!");
        }
```

(continues on next page)

(continued from previous page)

```
        String ifname = itf.GetString("-in");
        String ofname = itf.GetString("-out");

        oemolistream ifs = new oemolistream(ifname);
        OEGraphMol mol = new OEGraphMol();
        oechem.OEReadMolecule(ifs, mol);
        ifs.close();
        oedepict.OEPrepareDepiction(mol);

        double width  = oedepict.OEGetImageWidth (itf);
        double height = oedepict.OEGetImageHeight(itf);
        double scale  = OEScale.AutoScale;
        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
        // set up display options from command line parameters
        oedepict.OESetup2DMolDisplayOptions(opts, itf);

        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
        oedepict.OERenderMolecule(ofname, disp);
        return;
    }
}
```

```
prompt> java DepictMolConfigure --help
```

will generate the following standard interface:

```
Simple parameter list
    image options
      -height : Height of output image
      -width : Width of output image

    input/output
      -in : Input filename
      -out : Output filename

    molecule display options
      -aromstyle : Aromatic ring display style
      -atomcolor : Atom coloring style
      -atomprop : Atom property display
      -atomstereostyle : Atom stereo display style
      -bondcolor : Bond coloring style
      -bondprop : Bond property display
      -bondstereostyle : Bond stereo display style
      -hydrstyle : Hydrogen display style
      -linewidth : Default bond line width
      -protgroupdisp : Protective group display style
      -scale : Scaling of the depicted molecule
      -superdisp : Super atom display style
      -titleloc : Location of the molecule title
```

See also

- OEInterface class in the **OEChem TK** manual
- *OEDepict Examples* chapter

OEDepict TK provides the following high-level functions to build standard interface of depiction applications:

Function name	Builds standard interface for ..
`OEConfigure2DMolDisplayOptions`	molecule depiction style
`OEConfigureHighlightParams`	highlighting
`OEConfigureImageGridParams`	image grid
`OEConfigureImageOptions`	image
`OEConfigureMultiPageParams`	multi-page image
`OEConfigurePrepareDepictionOptions`	preparing molecules for 2D depiction
`OEConfigureReportOptions`	multi-page report

2.3 Highlighting

Atoms and bonds of a molecule can be highlighted by using the `OEAddHighlighting` functions. **OEDepict TK** provides four built-in highlighting styles:

color
: This style is associated with the `OEHighlightStyle.Color` constant. See *Figure: Example of highlighting using the 'color' style*.

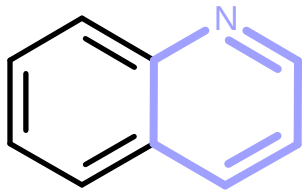


Fig. 14: **Example of highlighting using the 'color' style**

stick
: This style is associated with the `OEHighlightStyle.Stick` constant. See *Figure: Example of highlighting using the 'stick' style*.

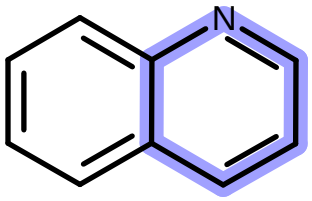


Fig. 15: **Example of highlighting using the 'stick' style**

ball and stick
: This style is associated with the `OEHighlightStyle.BallAndStick` constant. See *Figure: Example of highlighting using the 'ball and stick' style*.

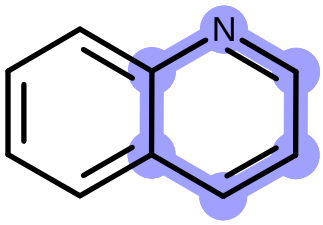


Fig. 16: **Example of highlighting using the 'ball and stick' style**

cogwheel
: This style is associated with the `OEHighlightStyle.Cogwheel` constant. See *Figure: Example of highlighting using the 'cogwheel' style*.

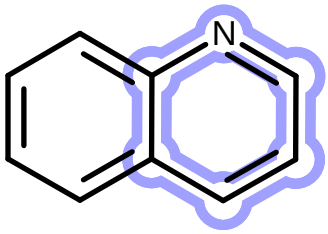


Fig. 17: **Example of highlighting using the 'cogwheel' style**

lasso
: This style is associated with the `OEHighlightStyle.Lasso` constant. See *Figure: Example of highlighting using the 'lasso' style*.

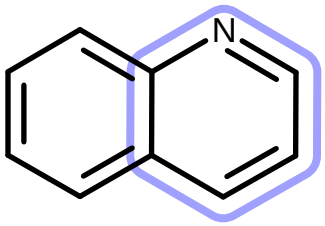


Fig. 18: **Example of highlighting using the 'lasso' style**

2.3.1 Using Highlighting Styles

The following example (*Listing 1*) shows how to use the built-in highlighting styles. The *OEAddHighlighting* function takes an OEMatchBase object and highlights each of its target atoms and bonds using the given style. In this case, the two pyridine rings in the target structure are highlighted by red color using the **stick** style. The image created by *Listing 1* is shown in *Figure: Example of using 'stick' highlighting style*.

Listing 1: Example of using highlighting style

```
public class HighlightSimple {
    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cncc2c1cc3c(cncc3)c2");
        oedepict.OEPrepareDepiction(mol);

        OESubSearch ss = new OESubSearch("c1ncccc1");

        OE2DMolDisplay disp = new OE2DMolDisplay(mol);

        boolean unique = true;
        for (OEMatchBase match : ss.Match(mol, unique)) {
            oedepict.OEAddHighlighting(disp, oechem.getOERed(), OEHighlightStyle.Stick,␣
↪match);
        }

        oedepict.OERenderMolecule("HighlightSimple.svg", disp);
    }
}
```

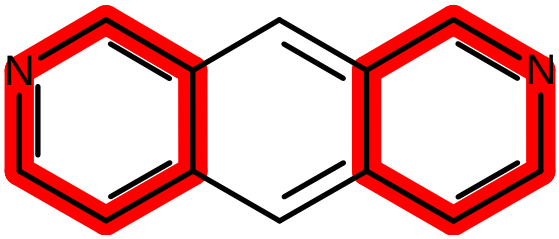


Fig. 19: **Example of using 'stick' highlighting style**

See also

- *OEHighlightStyle* namespace
- *OEAddHighlighting* function
- *Depicting Substructure Search Match* example

2.3.2 Using Highlighting Classes

OEDepict TK also provides the highlighting classes that correspond to the highlighting styles. Each highlighting style can be customized by using the corresponding class:

- *OEHighlightByColor*
- *OEHighlightByStick*
- *OEHighlightByBallAndStick*
- *OEHighlightByCogwheel*
- *OEHighlightByLasso*

The next example (*Listing 2*) shows how the **stick** style can be customized by using the corresponding *OEHighlightByStick* class. The image created by *Listing 2* is shown in *Figure: Example of customizing the 'stick' highlighting style*.

Listing 2: Example of using highlighting class

```
public class HighlightStyle {
    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cncc2c1cc3c(cncc3)c2");
        oedepict.OEPrepareDepiction(mol);

        OESubSearch ss =  new OESubSearch("c1ncccc1");

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(240.0, 100.0, OEScale.
↪AutoScale);
        opts.SetMargins(10.0);
        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);

        double stickWidthScale = 6.0;
        boolean monochrome = true;
        OEHighlightByStick highlight = new OEHighlightByStick(oechem.getOERed(),
                stickWidthScale,
                !monochrome);

        boolean unique = true;
        for (OEMatchBase match : ss.Match(mol, unique)) {
            oedepict.OEAddHighlighting(disp, highlight, match);
        }

        oedepict.OERenderMolecule("HighlightStyle.svg", disp);
    }
}
```

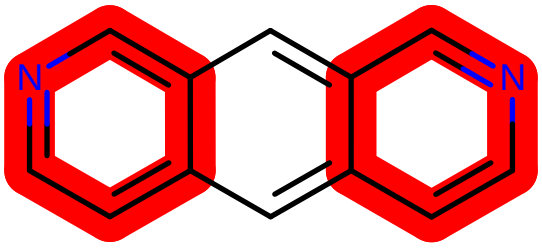


Fig. 20: **Example of customizing the 'stick' highlighting style**

See also

- *OEAddHighlighting* function

2.3.3 Highlighting Overlapped Patterns

If more than one part of a molecule is highlighted, these highlights can overlap resulting in loss of information. For example, highlighting the matches of the *c1cc[c,n]cc1* SMARTS pattern will produce the following image.

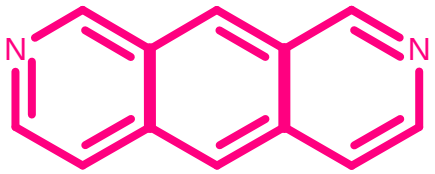


Fig. 21: **Example of highlighting multiple matches in the same image**

There are two ways how this can be avoided and correctly highlight overlapping parts of a molecule. The first example demonstrates how to depict multiple matches in different image cells one by one. The image created by *Listing 3* is shown in *Figure: Example of highlighting multiple matches one by one*.

Listing 3: Example of highlighting multiple matches one by one

```
public class HighlightMulti {
    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cncc2c1cc3c(cncc3)c2");
        oedepict.OEPrepareDepiction(mol);

        OESubSearch ss = new OESubSearch("c1cc[c,n]cc1");

        double width  = 350;
        double height = 250;
        OEImage image = new OEImage(width, height);
```

(continues on next page)

(continued from previous page)

```
        int rows = 2;
        int cols = 2;
        OEImageGrid grid = new OEImageGrid(image, rows, cols);
        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(grid.GetCellWidth(),
                grid.GetCellHeight(),
                OEScale.AutoScale);

        boolean unique = true;
        OEImageBaseIter celliter = grid.GetCells();
        for (OEMatchBase match : ss.Match(mol, unique)) {
            OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
            oedepict.OEAddHighlighting(disp, oechem.getOEPink(), OEHighlightStyle.Color,␣
↪match);
            oedepict.OERenderMolecule(celliter.Target(), disp);
            celliter.Increment();
        }
        oedepict.OEWriteImage("HighlightMulti.svg", image);
    }
}
```

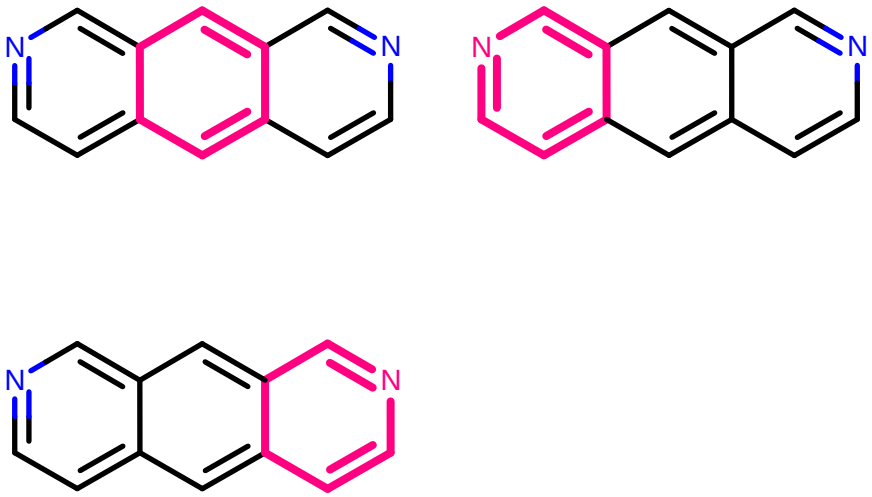


Fig. 22: **Example of highlighting multiple matches one by one**

See also

- *OEImageGrid* class
- *Depicting Molecules in a Grid* section

The second example (`Listing 4`) illustrates how to highlight overlapping matches at once using the *OEHighlightOver-*

layByBallAndStick class. In this example, the *OEAddHighlightOverlay* function takes all matches being highlighted and colors the overlapped atoms and bonds using the colors by turn. The colors used for highlighting are determined when the *OEHighlightOverlayByBallAndStick* object is constructed. The image created by *Listing 4* is shown in *Figure: Example of highlighting overlapping matches at once*.

Listing 4: Example of highlighting multiple matches simultaneously

```
public class HighlightOverlay {
    public static void main(String argv[]) {
        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "c1cncc2c1cc3c(cncc3)c2");
        oedepict.OEPrepareDepiction(mol);

        OESubSearch subs = new OESubSearch("c1cc[c,n]cc1");

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(240.0, 100.0, OEScale.
↪AutoScale);
        opts.SetMargins(10.0);
        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);

        OEHighlightOverlayByBallAndStick highlight =
            new OEHighlightOverlayByBallAndStick(oechem.OEGetContrastColors());

        boolean unique = true;
        oedepict.OEAddHighlightOverlay(disp, highlight, subs.Match(mol, unique));
        oedepict.OERenderMolecule("HighlightOverlay.png", disp);
    }
}
```

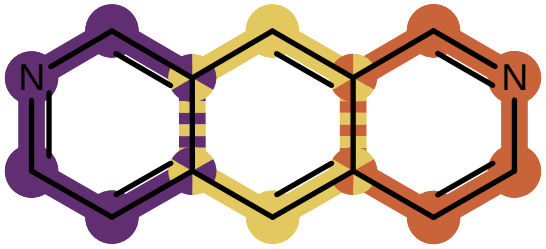


Fig. 23: **Example of highlighting overlapping matches at once**

See also

- *OEHighlightOverlayBase* base class
- *OEHighlightOverlayByBallAndStick* class
- *OEHighlightOverlayStyle* namespace
- OEGetColors, OEGetContrastColors, OEGetDeepColors, OEGetLightColors and OEGetVividColors functions

2.4 Molecule Layouts

The previous chapters described how to depict a single molecule in an image. This chapter provides examples for drawing multiple molecular diagrams into the same image.

2.4.1 Depicting Molecules in an Arbitrary Position

The *OEImageFrame* class provides a general framework to display a molecule in a specific region of the image by defining the following parameters:

- the reference of the *parent* image
- the width and height of the generated *frame*
- the offset of the *frame* relative to the top-left corner of its *parent* (*OE2DPoint*)

The *Listing 1* examples generates an image where the same molecule is depicted in different regions of the image. The molecular diagrams are automatically resized in order to fit into the specific frame, since the corresponding *OE2DMolDisplay* object is constructed with the width and height of the *OEImageFrame* object. When the *OERenderMolecule* function is called, the offset of the *OEImageFrame* object is added to the coordinates stored in *OE2DMolDisplay* in order to transfer the molecule into the specific region of the image. The image created by *Listing 1* is shown in *Figure: Example of using OEImageFrame*.

Listing 1: Example of using OEImageFrame

```
public class ImageFrame {
    private static void DrawMolecule(OEImageBase image, OEMolBase mol,
            double width, double height, OE2DPoint offset) {
        OEImageFrame frame = new OEImageFrame(image, width, height, offset);

        oedepict.OEDrawBorder(frame, oedepict.getOELightGreyPen());

        double scale = OEScale.AutoScale;
        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
        OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
        boolean clearBackground = true;
        oedepict.OERenderMolecule(frame, disp, !clearBackground);
    }

    public static void main(String argv[]) {
        OEImage image = new OEImage(400, 400);

        OEGraphMol mol = new OEGraphMol();
        oechem.OESmilesToMol(mol, "C1CC(O)CNC1");
```

(continues on next page)

(continued from previous page)

```
        oedepict.OEPrepareDepiction(mol);

        DrawMolecule(image, mol,  60,  60, new OE2DPoint(50.0,   40.0));
        DrawMolecule(image, mol, 180, 180, new OE2DPoint(180.0, 120.0));
        DrawMolecule(image, mol,  80,  80, new OE2DPoint(300.0,  20.0));
        DrawMolecule(image, mol,  50,  50, new OE2DPoint(150.0, 320.0));
        DrawMolecule(image, mol,  20,  20, new OE2DPoint(360.0, 360.0));

        oedepict.OEWriteImage("ImageFrame.png", image);
    }
}
```

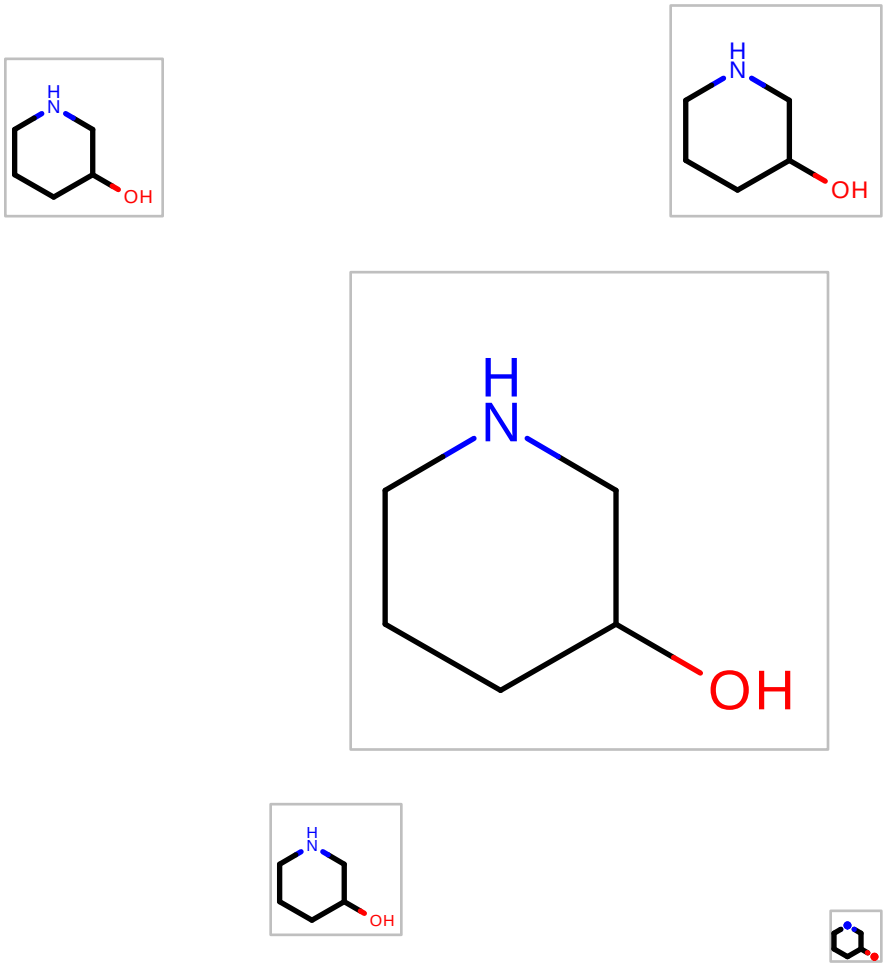


Fig. 24: **Example of using OEImageFrame**

See also

- *OEImageFrame* class
- *OEDrawBorder* function
- *OERenderMolecule* function
- *OEWriteImage* function

2.4.2 Depicting Molecules in a Grid

The *OEImageGrid* class allows molecules to be aligned into rows and columns. After generating an *OEImageGrid* object by specifying its number of rows and columns, the cells of the grid can be accessed (from left to right, top to bottom order) by calling the *OEImageGrid.GetCells* method. The image created by *Listing 2* is shown in *Figure: Example of using OEImageGrid*.

Listing 2: Example of using OEImageGrid

```
public class ImageGridSimple {
    public static void main(String args[]) {
        ArrayList<String> smiles = new ArrayList<String>();
        smiles.add("C1CC(C)CCC1");
        smiles.add("C1CC(O)CCC1");
        smiles.add("C1CC(Cl)CCC1");

        OEImage image = new OEImage(200, 200);

        int rows = 2;
        int cols = 2;
        OEImageGrid grid = new OEImageGrid(image, rows, cols);

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(grid.GetCellWidth(),
                grid.GetCellHeight(),
                OEScale.AutoScale);
        OEImageBaseIter celliter = grid.GetCells();
        for (String smi : smiles) {
            OEGraphMol mol = new OEGraphMol();
            oechem.OESmilesToMol(mol, smi);
            oedepict.OEPrepareDepiction(mol);

            OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
            oedepict.OERenderMolecule(celliter.Target(), disp);
            celliter.Increment();
        }

        oedepict.OEWriteImage("ImageGridSimple.svg", image);
    }
}
```

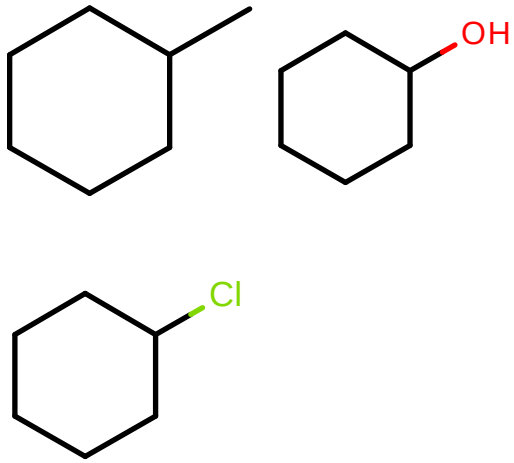


Fig. 25: **Example of using OEImageGrid**

See also

- *OEImageGrid* class

When an *OE2DMolDisplay* object is initialized, the dimensions of the depicted molecule is controlled by the width, height and scale parameters of the *OE2DMolDisplayOptions* object.

When only one molecule is depicted then auto scaling of this molecule is preferred (i.e. scaling the molecule in order to maximally fit it to the given dimensions). However, when more that one molecule, that are quite different in sizes, are depicted next to each other, auto scaling of each molecule can be visually misleading because it emphasizes molecules with greater scaling factors. See example in *Figure: Example of using OEImageGrid* that is generated by `Listing 3`.

Listing 3: Example of auto scaling of molecules

```
public class ImageGridAutoScale {

    public static void main(String[] args) {
        ArrayList<String> smiles = new ArrayList<String>();
        smiles.add("c1ccccc1");
        smiles.add("c1cccc2c1ccnc2");
        smiles.add("c1ccc2c(c1)cc3ccccc3n2");

        OEImage image = new OEImage(400, 200);

        int rows = 1;
        int cols = 3;
        OEImageGrid grid = new OEImageGrid(image, rows, cols);

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(grid.GetCellWidth(),
                grid.GetCellHeight(),
                OEScale.AutoScale);
```

(continues on next page)

(continued from previous page)

```
        int counter = 0;
        for (OEImageBase cell : grid.GetCells()) {
            OEGraphMol mol = new OEGraphMol();
            oechem.OESmilesToMol(mol, smiles.get(counter));
            oedepict.OEPrepareDepiction(mol);

            OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
            oedepict.OERenderMolecule(cell, disp);
            counter += 1;
        }

        oedepict.OEWriteImage("ImageGridAutoScale.svg", image);
    }
}
```

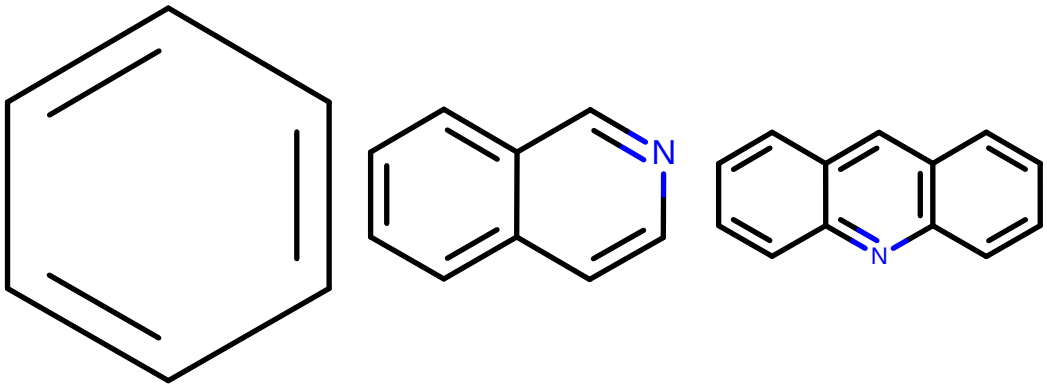


Fig. 26: **Example of auto scaling of molecules**

See also

Controlling the Size of Depicted Molecules section

The following example (*Listing 4*) shows how to ensure that all molecules are depicted using the same scaling factor. After initializing the molecules, the minimum scaling factor is determined by looping over the molecules being depicted and calling the *OEGetMoleculeScale* function. This function returns the scaling factor of the molecule considering the depiction options stored in the given *OE2DMolDisplayOptions* object. After determining the minimum scaling factor, the molecules are rendered to the cells of the grid by using this fixed scaling factor. The image created by *Listing 4* is shown in *Figure: Example of using OEImageGrid*.

Listing 4: Example of identical scaling of molecules

```
public class ImageGridFixedScale {
    public static void main(String[] args) {
        ArrayList<String> smiles = new ArrayList<String>();
        smiles.add("c1ccccc1");
        smiles.add("c1cccc2c1ccnc2");
        smiles.add("c1ccc2c(c1)cc3ccccc3n2");

        ArrayList<OEGraphMol> mollist = new ArrayList<OEGraphMol>();
        for (String smi : smiles) {
            OEGraphMol mol = new OEGraphMol();
            oechem.OESmilesToMol(mol, smi);
            oedepict.OEPrepareDepiction(mol);
            mollist.add(mol);
        }

        OEImage image = new OEImage(400, 200);

        int rows = 1;
        int cols = 3;
        OEImageGrid grid = new OEImageGrid(image, rows, cols);

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(grid.GetCellWidth(),
                grid.GetCellHeight(),
                OEScale.AutoScale);

        double minscale = Double.MAX_VALUE;
        for (OEGraphMol mol : mollist) {
            minscale = Math.min(minscale, oedepict.OEGetMoleculeScale(mol, opts));
        }
        opts.SetScale(minscale);

        OEImageBaseIter celliter = grid.GetCells();
        for (OEGraphMol mol : mollist) {
            OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
            oedepict.OERenderMolecule(celliter.Target(), disp);
            celliter.Increment();
        }

        oedepict.OEWriteImage("ImageGridFixedScale.svg", image);
```

(continues on next page)

(continued from previous page)

```
    }
}
```

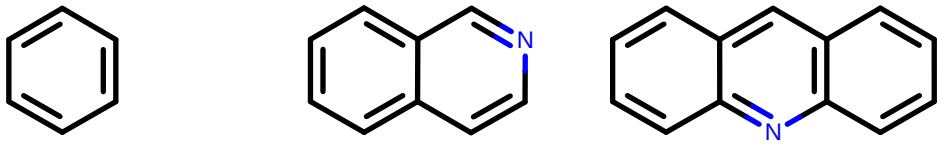


Fig. 27: **Example of identical scaling of molecules**

See also

- *OEImageGrid.GetCellHeight* method
- *OEImageGrid.GetCellWidth* method

2.4.3 Depicting Molecules in a Table

The following example demonstrates how to depict molecules and display data alongside them by using the *OEImageTable*. The individual cells of an *OEImageTable* object can be accessed by calling the any of the following methods:

- *OEImageTable.GetCell*
- *OEImageTable.GetBodyCell*
- *OEImageTable.GetHeaderCell*
- *OEImageTable.GetStubColumnCell*

These methods return an image (*OEImageBase*) on which molecule can be drawn or text can be displayed using the *OEImageTable.DrawText* method. The image created by *Listing 5* is shown in *Figure: Example of depicting molecules and data using image table*.

Listing 5: Example of depicting molecules and data using image table

```
        ArrayList<String> smiles = new ArrayList<String>();
        smiles.add("c1ccccc1 benzene");
        smiles.add("c1cccnc1 pyridine");
        smiles.add("c1cncnc1 pyrimidine");
```

(continues on next page)

(continued from previous page)

```
        int imagewidth = 350;
        int imageheight = 250;
        OEImage image = new OEImage(imagewidth, imageheight);

        int mainrows = 3;
        int maincols = 2;
        OEImageTableOptions maintableopts = new OEImageTableOptions(mainrows, maincols,␣
↪OEImageTableStyle.NoStyle);
        maintableopts.SetHeader(false);
        ArrayList<Integer> colwidths = new ArrayList<Integer>();
        colwidths.add(10);
        colwidths.add(20);
        maintableopts.SetColumnWidths(colwidths);
        OEImageTable maintable = new OEImageTable(image, maintableopts);

        int datarows = 4;
        int datacols = 2;
        OEImageTableOptions datatableopts = new OEImageTableOptions(datarows, datacols,␣
↪OEImageTableStyle.LightGreen);
        datatableopts.SetStubColumn(true);
        datatableopts.SetMargins(5.0);
        datatableopts.SetColumnWidths(colwidths);

        for (int r = 0; r < mainrows; ++r) {
            // depict molecule in first column
            OEImageBase cell = maintable.GetBodyCell(r + 1, 1);
            OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(cell.GetWidth(), cell.
↪GetHeight(), OEScale.AutoScale);
            opts.SetTitleLocation(OETitleLocation.Hidden);

            OEGraphMol mol = new OEGraphMol();
            oechem.OESmilesToMol(mol, smiles.get(r));
            oedepict.OEPrepareDepiction(mol);

            OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
            oedepict.OERenderMolecule(cell, disp);

            // depicting data in table
            cell = maintable.GetBodyCell(r + 1, 2);
            OEImageTable datatable = new OEImageTable(cell, datatableopts);

            datatable.DrawText(datatable.GetHeaderCell(1, false), "Property");
            datatable.DrawText(datatable.GetHeaderCell(1), "Value");

            datatable.DrawText(datatable.GetStubColumnCell(1), "Name");
            datatable.DrawText(datatable.GetBodyCell(1, 1), mol.GetTitle());

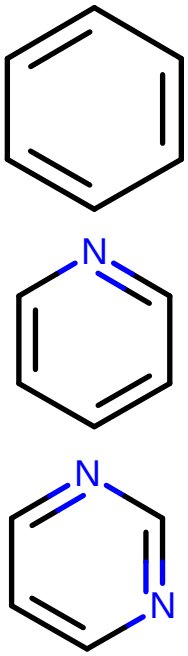
            datatable.DrawText(datatable.GetStubColumnCell(2), "SMILES");
            datatable.DrawText(datatable.GetBodyCell(2, 1), oechem.OEMolToSmiles(mol));

            datatable.DrawText(datatable.GetStubColumnCell(3), "MV");
            String mwdata = String.format("%.3f", oechem.
```

(continues on next page)

(continued from previous page)

```
↪OECalculateMolecularWeight(mol));
            datatable.DrawText(datatable.GetBodyCell(3, 1), mwdata);
        }
        oedepict.OEWriteImage("ImageTable.png", image);
```



Property	Value
Name	benzene
SMILES	c1ccccc1
MV	78.112

Property	Value
Name	pyridine
SMILES	c1ccncc1
MV	79.100

Property	Value
Name	pyrimidine
SMILES	c1cncnc1
MV	80.088

Fig. 28: **Example of depicting molecules and data using image table**

2.5 Multi Page Images

Multi-page images can be generated by the usage of the *OEMultiPageImageFile* class. After generating a multi-page object by specifying the orientation and the size of its pages, the individual pages can be accessed by calling the *OEMultiPageImageFile.NewPage* method. The returned *OEImage* then can be used to depict molecule(s) as demonstrated in the previous chapters. The image created by *Listing 1* is shown in *Figure: Example multi-page depiction*.

Listing 1: Example of multi-page depiction

```
public class MultiPage {
    private static OEImageGrid MakeNewPageGrid(OEMultiPageImageFile multi) {

        int rows = 2;
        int cols = 2;

        OEImage image = multi.NewPage();
        OEImageGrid grid = new OEImageGrid(image, rows, cols);
        grid.SetCellGap(20);
        grid.SetMargins(20);
        return grid;
    }
```

(continues on next page)

(continued from previous page)

```
    public static void main(String argv[]) {
        ArrayList<String> smiles = new ArrayList<String>();
        smiles.add("C1CC(C)CCC1");
        smiles.add("C1CC(O)CCC1");
        smiles.add("C1CC(Cl)CCC1");
        smiles.add("C1CC(F)CCC1");
        smiles.add("C1CC(Br)CCC1");
        smiles.add("C1CC(N)CCC1");

        OEMultiPageImageFile multi = new OEMultiPageImageFile(OEPageOrientation.
↪Landscape,
                OEPageSize.US_Letter);

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions();

        OEImageGrid grid = MakeNewPageGrid(multi);
        OEImageBaseIter celliter = grid.GetCells();
        for (String smi : smiles) {
            if (!celliter.IsValid()) {
                // go to new page
                grid = MakeNewPageGrid(multi);
                celliter = grid.GetCells();
            }
            OEImageBase cell = celliter.Target();

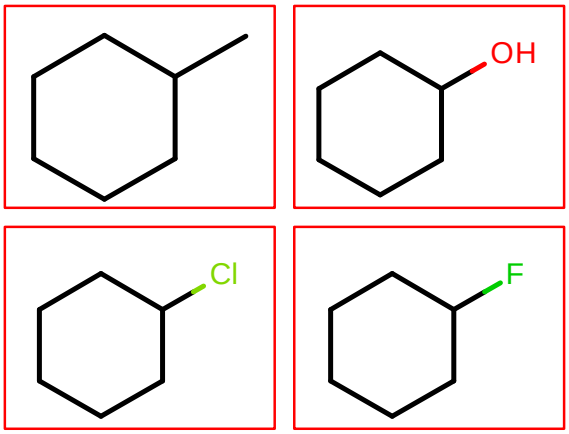
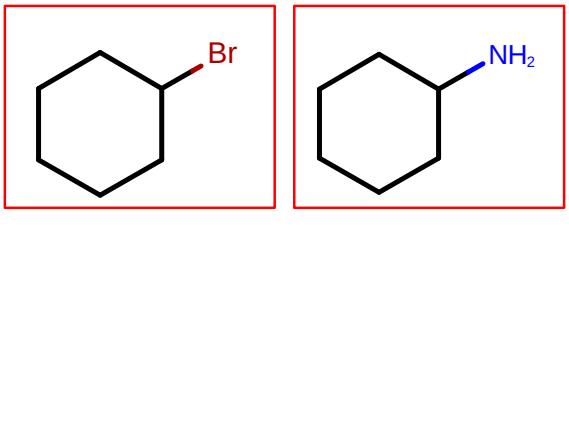
            OEGraphMol mol = new OEGraphMol();
            oechem.OESmilesToMol(mol, smi);
            oedepict.OEPrepareDepiction(mol);

            opts.SetDimensions(cell.GetWidth(), cell.GetHeight(), OEScale.AutoScale);
            OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
            oedepict.OERenderMolecule(cell, disp);
            oedepict.OEDrawBorder(cell, oedepict.getOERedPen());

            celliter.Increment();
        }

        oedepict.OEWriteMultiPageImage("MultiPage.pdf", multi);
    }
}
```

Table 3: **Example multi-page depiction** (The pages are reduced here for visualization convenience)

page 1	page 2
	

The *Listing 1* example creates a PDF multi-page image file, the multi-page image types supported by **OEDepict TK** can be accessed by calling the *OEGetSupportedMultiPageImageFileExtensions* function. It returns an iterator over the file extensions that can be used when writing a multi-page image file by the *OEWriteMultiPageImage* function.

2.5.1 Multi Page Reports

The *Listing 1* example illustrates how depict a set of molecules in a multi-page PDF using the *OEImageGrid* and *OEMultiPageImageFile* classes. **OEDepict TK** also provides the *OEReport* class to do this task in a more convenient way. The *OEReport* class is a multi-page layout manager that handles the page generation and the positioning of information (such as text and molecule depiction).

In the *Listing 2* example, first a *OEReportOptions* object is created that stores all properties that determine the layout of a documentation. After generating the *OEReport* object a molecule can be positioned on a cell that is returned by the *OEReport.NewCell* method. The *OEReport.NewCell* method creates cells from left to right and top to bottom order on each page. If no more cells are left on the page, then a new page is created before returning the first cell of this new page.

After the document is generated, *i.e* all molecule are depicted, the *OEWriteReport* function writes the multi-page documentation into into a file with the given name. The image created by the *Listing 2* example is the same as depicted in *Figure: Example multi-page depiction*.

Listing 2: Example of multi-page depiction using the OEReport class

```
public class MultiPageReport {

    public static void main(String argv[]) {
        ArrayList<String> smiles = new ArrayList<String>();
        smiles.add("C1CC(C)CCC1");
        smiles.add("C1CC(O)CCC1");
        smiles.add("C1CC(Cl)CCC1");
        smiles.add("C1CC(F)CCC1");
        smiles.add("C1CC(Br)CCC1");
        smiles.add("C1CC(N)CCC1");
```

(continues on next page)

(continued from previous page)

```
        int rows = 2;
        int cols = 2;
        OEReportOptions reportopts = new OEReportOptions(rows, cols);
        reportopts.SetPageOrientation(OEPageOrientation.Landscape);
        reportopts.SetCellGap(20);
        reportopts.SetPageMargins(20);
        OEReport report = new OEReport(reportopts);

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(report.GetCellWidth(),␣
↪report.GetCellHeight(), OEScale.AutoScale);

        for (String smi : smiles) {
            OEGraphMol mol = new OEGraphMol();
            oechem.OESmilesToMol(mol, smi);
            oedepict.OEPrepareDepiction(mol);

            OEImageBase cell = report.NewCell();
            OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
            oedepict.OERenderMolecule(cell, disp);
            oedepict.OEDrawBorder(cell, oedepict.getOERedPen());
        }

        oedepict.OEWriteReport("MultiPageReport.pdf", report);
    }
}
```

See also

- *OEReportOptions* class
- *OEReport* class

2.6 MDL Query Depiction

In order to depict an MDL query the query molecule has to be imported by calling the `OEReadMDLQueryFile` function. The rest of the process is identical to depicting a molecule. First the *OEPrepareDepiction* function has to be called to generate the 2D atom coordinates, then the image can be created by rendering the molecule (*OERenderMolecule*).

Listing 1: Example of MDL query depiction

```
        if (argv.length != 2) {
            oechem.OEThrow.Usage("DepictMDLQuery <mdlquery> <imagefile>");
        }
        oemolistream ifs = new oemolistream(argv[0]);
        OEGraphMol qmol = new OEGraphMol();
        oechem.OEReadMDLQueryFile(ifs, qmol);
        ifs.close();
        oedepict.OEPrepareDepiction(qmol);
        oedepict.OERenderMolecule(argv[1], qmol);
```

See also

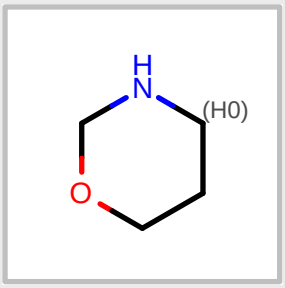
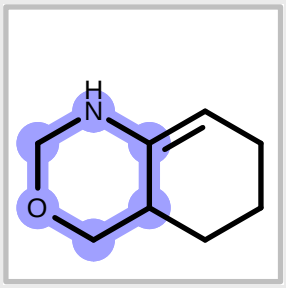
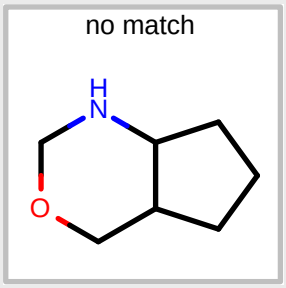
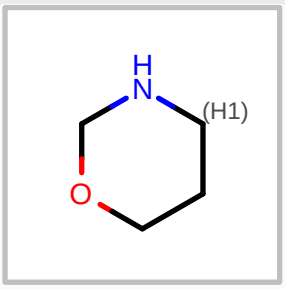
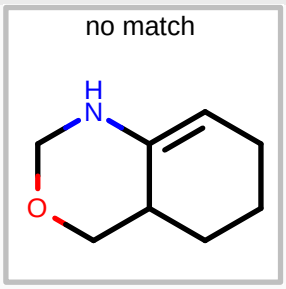
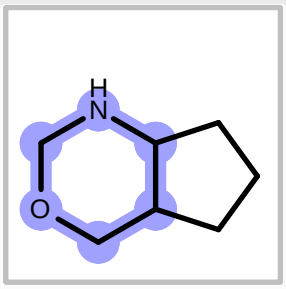
- `OEReadMDLQueryFile` function in the **OEChem TK** manual
- Table: Summary of supported MDL query features

2.6.1 Atom Query Features

The depiction of the following MDL atom query features is supported:

1. The 8th column in the atom block in the V2K format file (or `HCOUNT=` in the V3K format file) is used to define the number of allowed hydrogens for an atom. (See examples in *Table: Example of depicting hydrogen count*)
   - `(H0)` means no hydrogen atoms allowed unless explicitly drawn.
   - `(Hn)` means atom must have `n` or more implicit hydrogen(s)

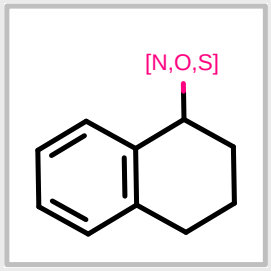
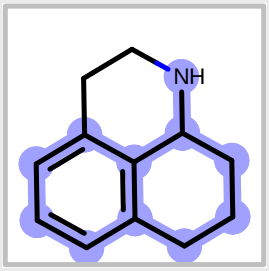
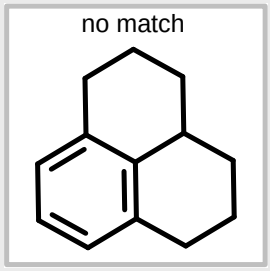
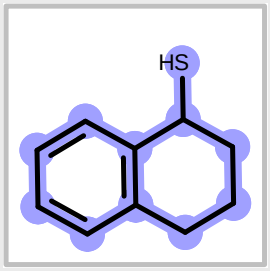
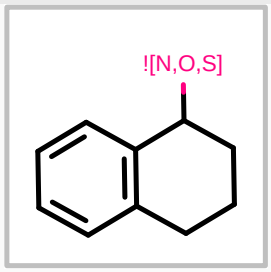
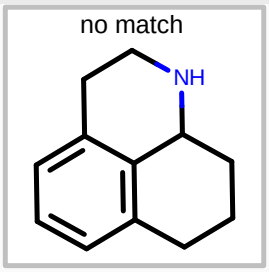
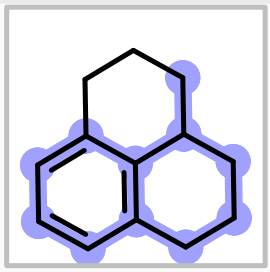
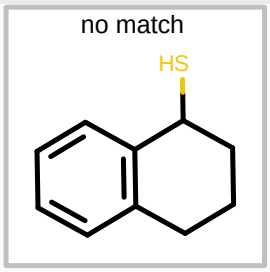
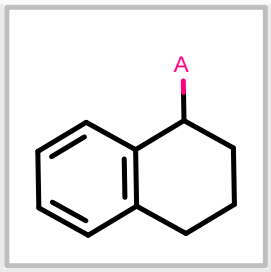
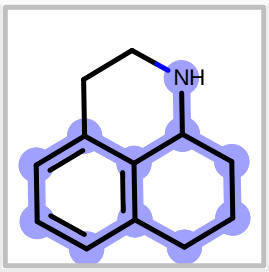
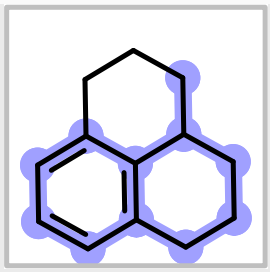
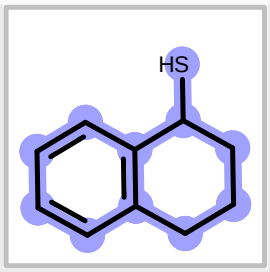
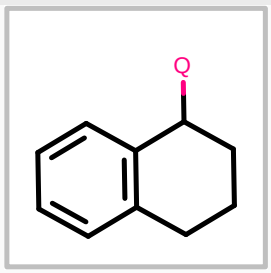
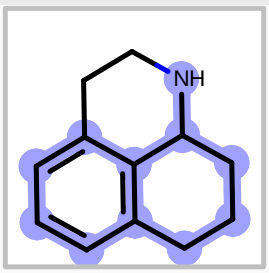
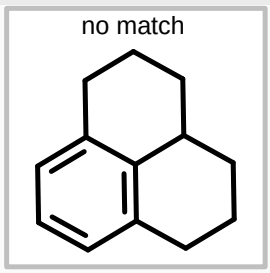
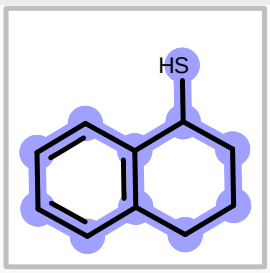
Table 4: **Examples of depicting hydrogen count**

query		
		
		

2. Query atom types:
   - `A` = any atom type except hydrogen
   - `Q` = any atom type except hydrogen and carbon
   - `L` = atom list

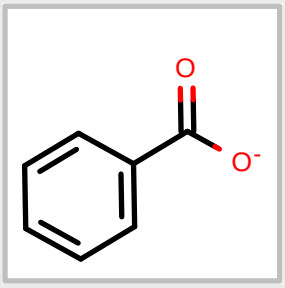
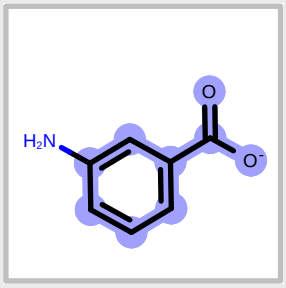
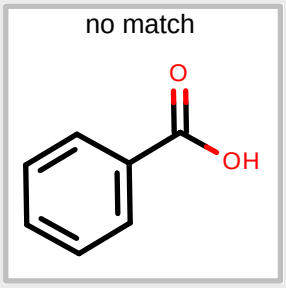
   The `M ALS` line in the property block in the V2K format file (or `[aaa,bbb,..]` in the V3K format) is used to list alternative atom types for an atom. (See examples in *Table: Examples of depicting query atom types*)

Table 5: **Examples of depicting query atom types**

query	target	target	target
			
			
			
			

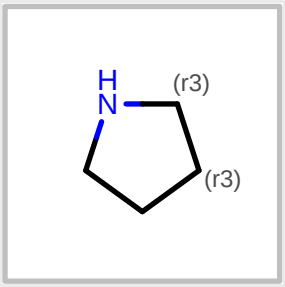
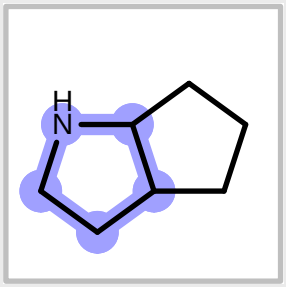
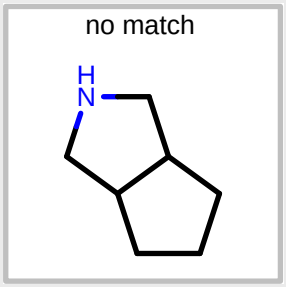
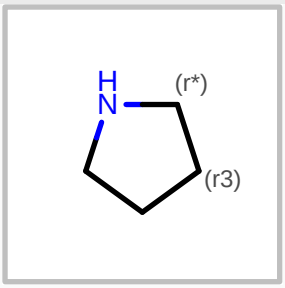
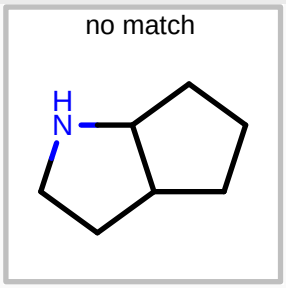
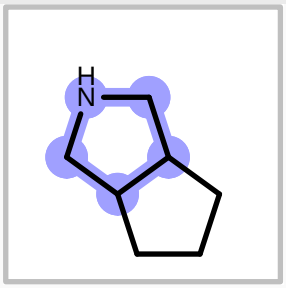
3. `M CHG` line in the property block in the V2K format file (or `CHG=` in the V3K format) is used to define atom formal charges. (See examples in *Table: Example of depicting formal charge*)

Table 6: **Examples of depicting formal charge**

query	target	target
		no match 

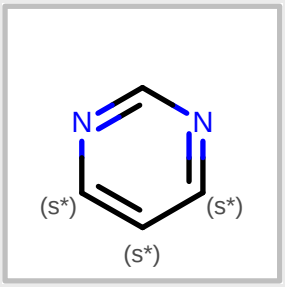
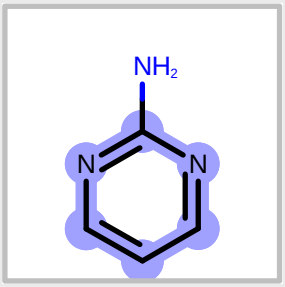
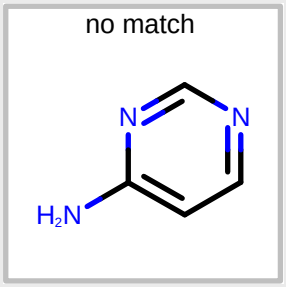
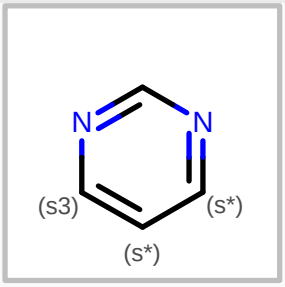
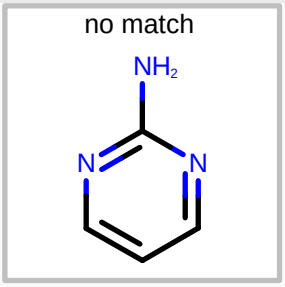
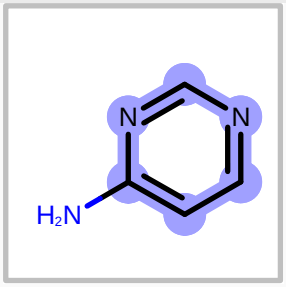
4. `M RBC` line in the property block in the V2K format file (or `RBCNT=` in the V3K format) is used to limit the number of allowed ring bonds attached to an atom. (See examples in *Table: Example of depicting ring count*)
   - `(r0)` means no ring bond allowed
   - `(r*)` means as drawn
   - `(rn)` means `n` number of ring bonds allowed

Table 7: **Examples of depicting ring count**

query	target	target
		no match 
	no match 	

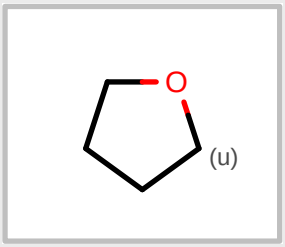
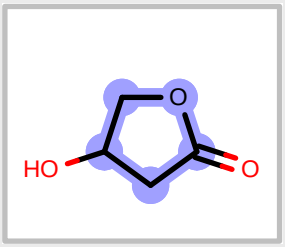
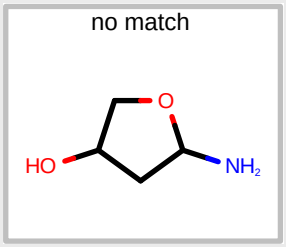
5. `M SUB` line in the property block in the V2K format file (or `SUBST=` in the V3K format) is used to set the number of allowed substitutions of an atom. (See examples in *Table: Examples of depicting substitution count*)
   - `(s0)` means no substitution allowed
   - `(s*)` means as drawn
   - `(sn)` means `n` number of substitution(s) allowed

Table 8: **Examples of depicting substitution count**

query	target	target
		
		

6. `M UNS` line in the property block in the V2K format file (or `UNSAT=` in the V3K format) is used to specify whether or not an atom is unsaturated, *i.e.*, having at least one multiple bond. (See examples in *Table: Examples of depicting unsaturated property*)

Table 9: **Examples of depicting unsaturated property**

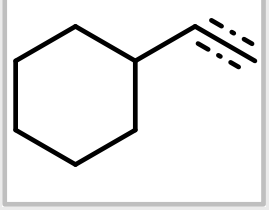
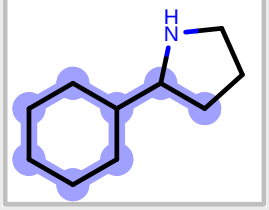
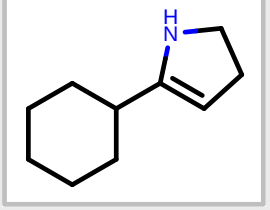
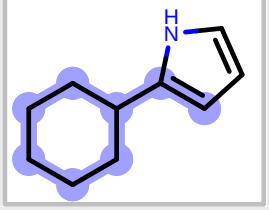
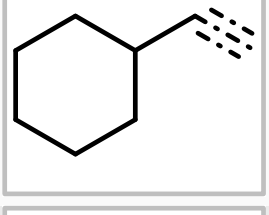
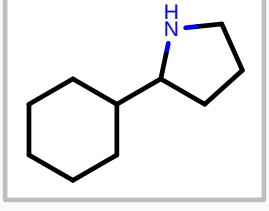
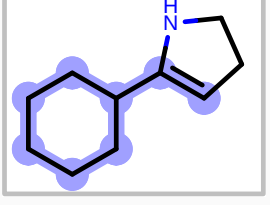
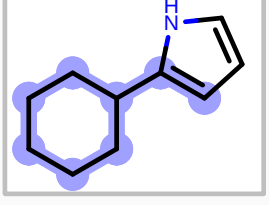
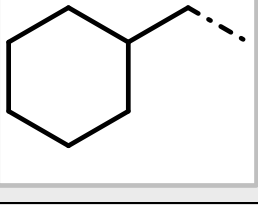
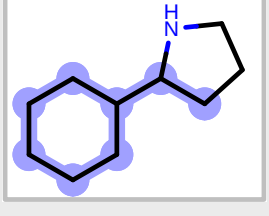
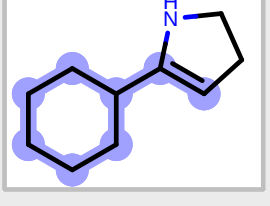
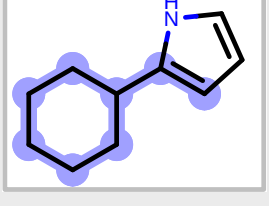
query	target	target
		

2.6.2 Bond Query Features

The depiction of the following MDL bond query features is supported:

1. Alternative bond types in the bond block (4 = aromatic, 5 = single or double, 6 = single or aromatic, 7 = double or aromatic, 8 = any bond). (See examples in *Table: Examples of depicting query bond types*)

Table 10: **Examples of depicting query bond types**

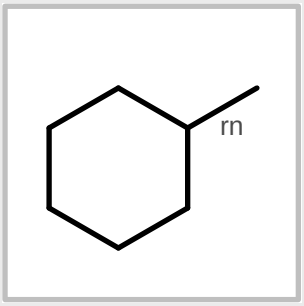
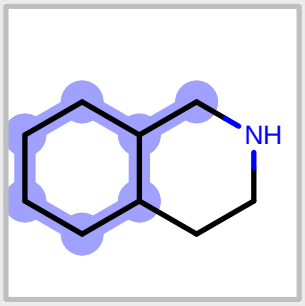
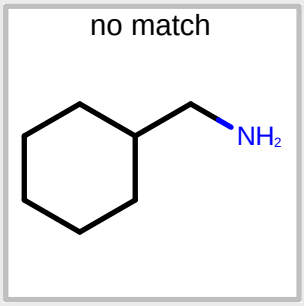
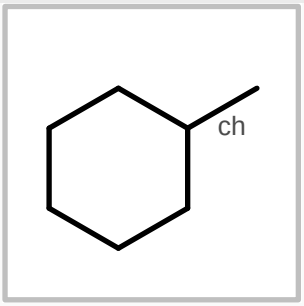
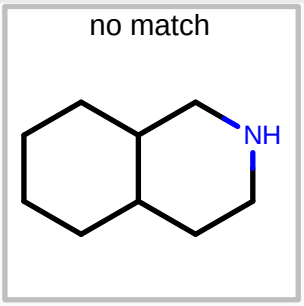
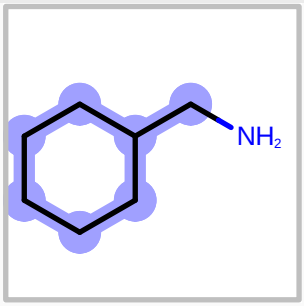
query	target	target	target
(single or aromatic) 		no match 	
(double or aromatic) 	no match 		
(any) 			

See also

- *OEBondDisplayType* namespace

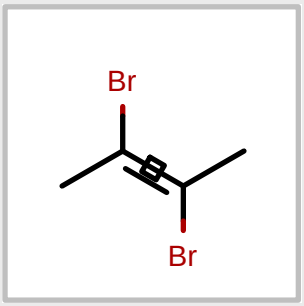
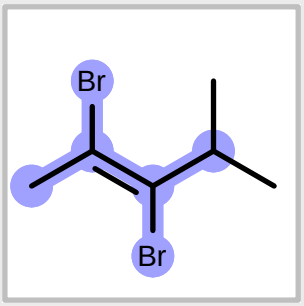
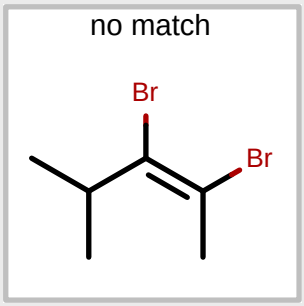
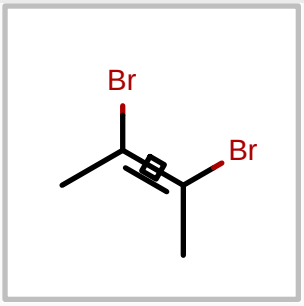
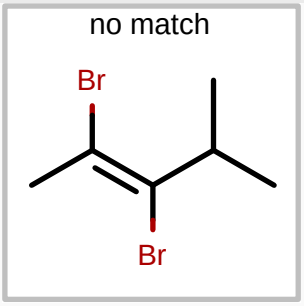
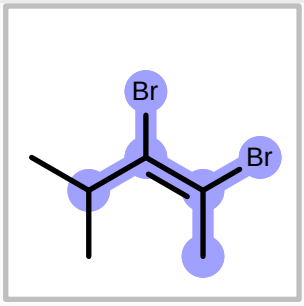
2. The 6th column in the atom block in the V2K format file (or TOPO= in the V3K format) describes bond topology. (See examples in *Table: Examples of depicting bond topology*)
   - (rn) means that it can only mapped to ring bond
   - (ch) means that it can only mapped to chain bond

Table 11: **Examples of depicting bond topology**

query	target	target
		
		

3. Double bond stereochemistry is considered if both ends of the bond are marked with *stereo care* flags in the atom block in the V2K format file (or `STBOX=` in the V3K format). (See examples in *Table: Examples of depicting bond stereo care*)

Table 12: **Examples of depicting bond stereo care**

query	target	target
		
		

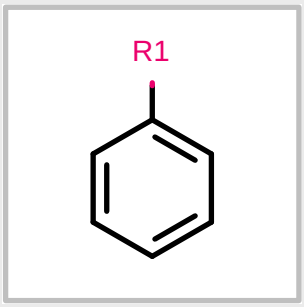
See also

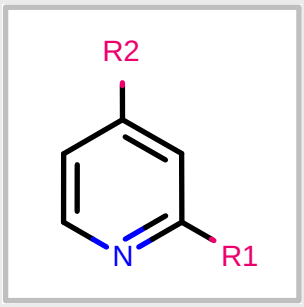
- *Depicting MDL Query* example

2.6.3 R-group Depiction

OEDepict TK can also depict R-group information by interpreting the `M RGP` line in the property block of an MDL file (See examples in *Table: Examples of depicting R-groups*)

Table 13: **Examples of depicting R-groups**





2.7 MDL Reaction Depiction

OEDepict TK also supports MDL reaction depiction. The reaction being depicted has to be imported by calling the `OEReadMDLReactionQueryFile` function. The rest of the process is identical to depicting a molecule. First the *`OEPrepareDepiction`* function has to be called to generate the 2D atom coordinates, then the image can be created by rendering the imported reaction (*`OERenderMolecule`*) (See examples in *Figure: Example of MDL reaction depiction*)

Listing 1: Example of reaction depiction

```
        if (argv.length != 2) {
            oechem.OEThrow.Usage("DepictMDLReaction <mdlreaction> <imagefile>");
        }
        oemolistream ifs = new oemolistream(argv[0]);
        OEGraphMol qmol = new OEGraphMol();
        oechem.OEReadMDLReactionQueryFile(ifs, qmol);
        ifs.close();
        oedepict.OEPrepareDepiction(qmol);
        oedepict.OERenderMolecule(argv[1], qmol);
    }
}
```

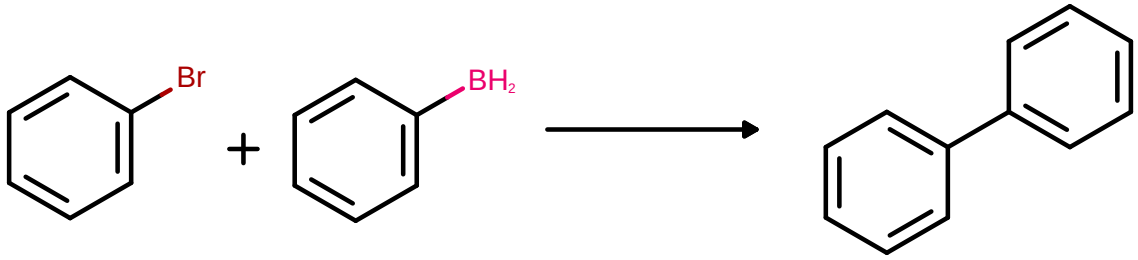


Fig. 29: **Example of MDL reaction depiction**

See also

- `OEReadMDLReactionQueryFile` function in the **OEChem TK** manual
- *Depicting MDL Reaction* example

The atom map information read from the reaction file can be depicted as atom properties by using the *OEDisplayAtomMapIdx* class. (See examples in *Figure: Example of MDL reaction depiction with atom mapping*)

Listing 2: Example of reaction depiction with map indices

```
        if (argv.length != 2) {
            oechem.OEThrow.Usage("DepictMDLReactionMapIdx <mdlreaction> <imagefile>");
        }

        String ifile = new String(argv[0]);
        String ofile = new String(argv[1]);

        oemolistream ifs = new oemolistream(ifile);
        OEGraphMol qmol = new OEGraphMol();
        oechem.OEReadMDLReactionQueryFile(ifs, qmol);
        ifs.close();
        oedepict.OEPrepareDepiction(qmol);

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions();
        opts.SetAtomPropertyFunctor(new OEDisplayAtomMapIdx());
        OE2DMolDisplay dist = new OE2DMolDisplay(qmol, opts);

        oedepict.OERenderMolecule(ofile, dist);
```

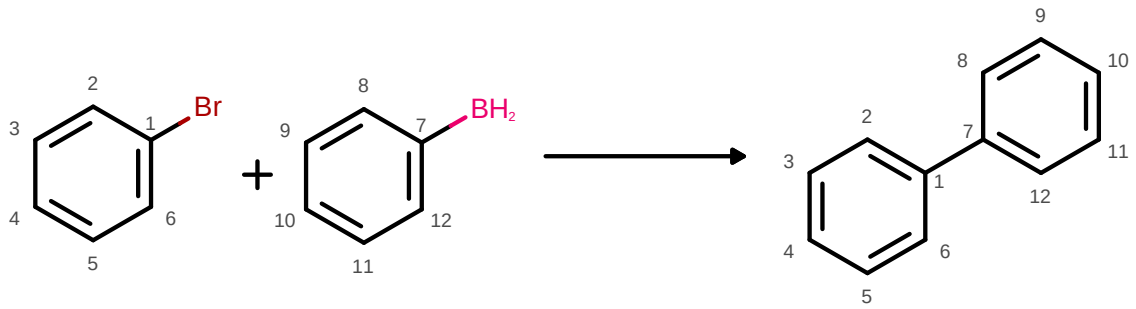


Fig. 30: **Example of MDL reaction depiction with atom mapping**

See also

- `OE2DMolDisplayOptions.SetAtomPropertyFunctor` method
- `Listing 5` example in the *Molecule Depiction* chapter

2.8 Molecule Alignment

The following three code examples demonstrate how to align molecules either based on:

- maximum common substructure (`Listing 1`),
- substructure matches (`Listing 2`), or
- molecular similarity (`Listing 3`).

2.8.1 Molecule Alignment Based on MCS

The *Figure: Example of depiction without alignment* is generated by highlighting the maximum common substructure of two molecules (MCS), but keeping the original orientation of the compounds.

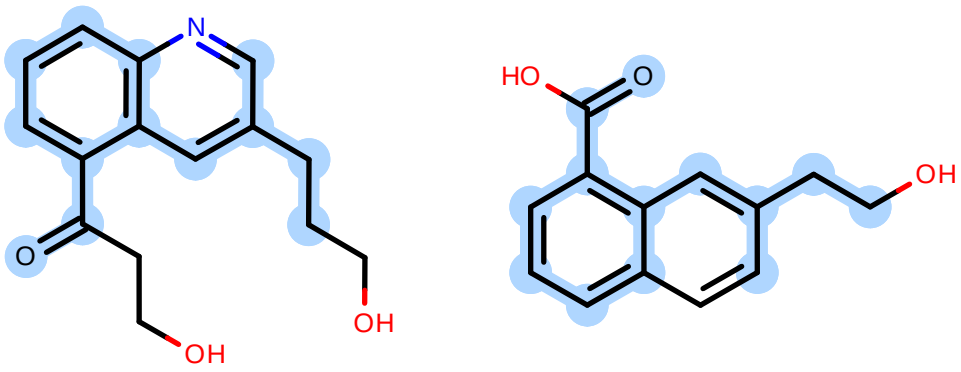


Fig. 31: **Example of depiction without alignment**

The *Listing 1* code example shows how to align these two molecules by their maximum common substructures (*i.e.* by the highlighted atoms and bonds in *Figure: Example of depiction without alignment*).

- First, two molecules are initialized from SMILES string and prepared for depiction.
- A OEMCSSearch object is initialized with the *reference* 2D molecule onto which the *fit* molecule will be aligned.
- An *OEImageGrid* object is constructed that allows to render the two molecules in a grid next to each other.
- The *fit* molecule is then aligned to the *reference* by calling the *OEPrepareAlignedDepiction* function that performs MCS search and then aligns the *fit* molecule to the *reference* based on the detected common substructure(s).
- An *OE2DMolDisplayOptions* object is initialized that stores the properties determine how the molecules are rendered. In order to render the molecules in equal size the smallest depiction scaling factor is calculated.
- If the molecule alignment was successful (*i.e* *OEAlignmentResult.IsValid* is *true*), then the correspondence between the two molecules stored in the *OEAlignmentResult* returned by the *OEPrepareAlignedDepiction* function. This *OEAlignmentResult* object can be used to highlight the matched common substructure by invoking the *OEAddHighlighting* function.
- After both molecules are rendered into the separate cells of the grid layout, the image is written into a `png` file.

After initializing the two molecules and the OEMCSSearch object, the `OEMCSSearch.Match` method is called to perform the search and return an iterator over the identified maximum common substructures. The first match is then utilized to call the *OEPrepareAlignedDepiction* function that aligns one molecule to the other based on the given match. After the alignment, the common atoms and bonds are highlighted by invoking the *OEAddHighlighting* function. The image created by *Listing 1* is shown in *Figure: Example of depiction with alignment based on MCS*.

Listing 1: Example of molecule alignment based on MCS

```
public class MCSAlign {
    public static void main(String argv[]) {
        OEGraphMol refmol = new OEGraphMol();
        oechem.OESmilesToMol(refmol, "c1cc(c2cc(cnc2c1)CCCO)C(=O)CCO");
        oedepict.OEPrepareDepiction(refmol);
```

(continues on next page)

(continued from previous page)

```
        OEGraphMol fitmol = new OEGraphMol();
        oechem.OESmilesToMol(fitmol, "c1cc2ccc(cc2c(c1)C(=O)O)CCO");
        oedepict.OEPrepareDepiction(fitmol);

        OEMCSSearch mcss = new OEMCSSearch(OEMCSType.Approximate);
        int atomexpr = OEExprOpts.DefaultAtoms;
        int bondexpr = OEExprOpts.DefaultBonds;
        mcss.Init(refmol, atomexpr, bondexpr);
        mcss.SetMCSFunc(new OEMCSMaxBondsCompleteCycles());

        OEAlignmentResult alignres = new OEAlignmentResult(oedepict.
↪OEPrepareAlignedDepiction(fitmol, mcss));

        OEImage image = new OEImage(400, 200);

        int rows = 1;
        int cols = 2;
        OEImageGrid grid = new OEImageGrid(image, rows, cols);

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(grid.GetCellWidth(),
                grid.GetCellHeight(),
                OEScale.AutoScale);
        opts.SetTitleLocation(OETitleLocation.Hidden);

        double refscale = oedepict.OEGetMoleculeScale(refmol, opts);
        double fitscale = oedepict.OEGetMoleculeScale(fitmol, opts);

        opts.SetScale(Math.min(refscale, fitscale));

        int hstyle = OEHighlightStyle.BallAndStick;

        OE2DMolDisplay refdisp = new OE2DMolDisplay(mcss.GetPattern(), opts);
        OE2DMolDisplay fitdisp = new OE2DMolDisplay(fitmol, opts);

        if (alignres.IsValid()) {
            OEAtomBondSet refabset = new OEAtomBondSet(alignres.GetPatternAtoms(),␣
↪alignres.GetPatternBonds());
            oedepict.OEAddHighlighting(refdisp, oechem.getOEBlueTint(), hstyle,␣
↪refabset);

            OEAtomBondSet fitabset = new OEAtomBondSet(alignres.GetTargetAtoms(),␣
↪alignres.GetTargetBonds());
            oedepict.OEAddHighlighting(fitdisp, oechem.getOEBlueTint(), hstyle,␣
↪fitabset);
        }

        OEImageBase refcell = grid.GetCell(1, 1);
        oedepict.OERenderMolecule(refcell, refdisp);

        OEImageBase fitcell = grid.GetCell(1, 2);
        oedepict.OERenderMolecule(fitcell, fitdisp);
```

(continues on next page)

(continued from previous page)

```
        oedepict.OEWriteImage("MCSAlign.png", image);
    }
}
```

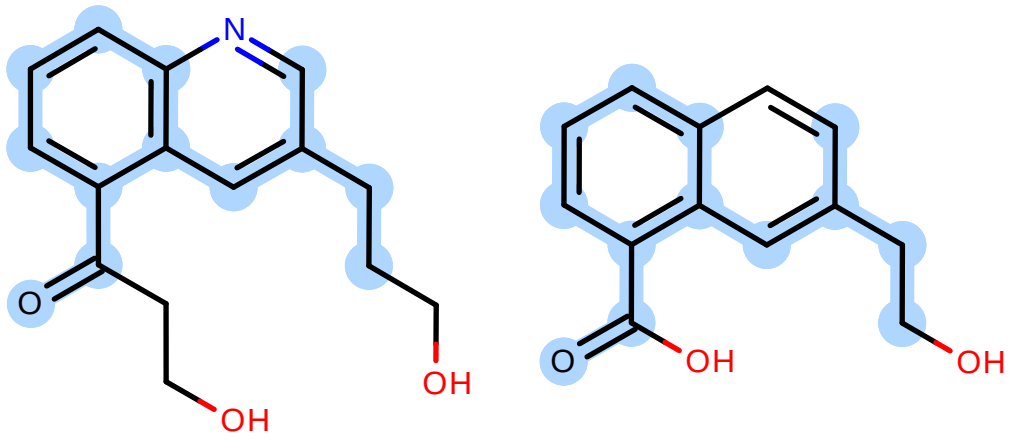


Fig. 32: **Example of depiction with alignment based on maximum common substructure**

See also

- OEMCSSearch class in the **OEChem TK** manual
- *OEImageGrid* class
- *OEAddHighlighting* function
- *OERenderMolecule* function
- *OEWriteImage* function
- *Aligning Molecule Based on MCS* example

2.8.2 Molecule Alignment Based on Substructure Search

The *Figure: Example of depiction without alignment* is generated by highlighting the "O=C(O)C(N)" substructures in four amino acids, but keeping the original orientation of the compounds.

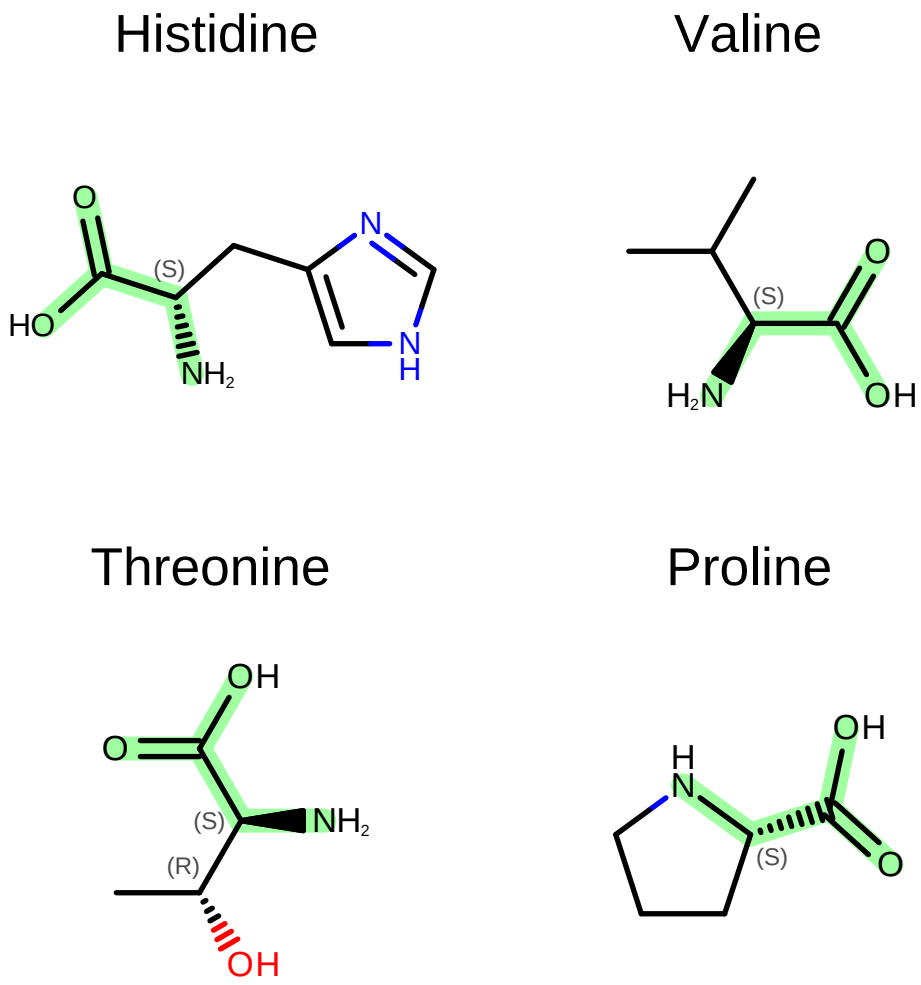


Fig. 33: **Example of depiction without alignment**

The *Listing 2* code example shows how to align these molecules by their matched substructures (*i.e.* by the highlighted atoms and bonds in *Figure: Example of depiction without alignment*).

- First, the OESubSearch object is initialized with the *reference* 2D molecule onto which all amino acid structures will be aligned.
- Each amino acids structure is initialized from a SMILES string and prepared for depiction.
- An *OEImageGrid* object is constructed that allows to render the molecules into a grid layout.
- An *OE2DMolDisplayOptions* object is initialized that stores the properties determine how the molecules are rendered. In order to render the molecules in equal size the smallest depiction scaling factor is calculated.
- Looping over the amino acids:
  - Each molecule is aligned to the reference molecule by calling the `OEPrepareAlignedDepiction` function that performs the alignment based on the detected substructure search match(es).
  - If the molecule alignment was successful (*i.e* `OEAlignmentResult.IsValid` is *true*), then the correspondence between the two molecules stored in the *OEAlignmentResult* returned by the

OEPrepareAlignedDepiction function. This *OEAlignmentResult* object can be used to highlight the matched substructure by invoking the *OEAddHighlighting* function.

- The aligned (and highlighted) molecule is then rendered to a next cell of the *OEImageGrid* object.

- Finally, the image is written into a png file.

The image created by *Listing 2* is shown in *Figure: Example of depiction with alignment based on substructure search*.

Listing 2: Example of molecule alignment based on substructure search

```
public class SubSearchAlign {
    public static void main(String argv[]) {
        String refsmiles = "O=C(O)C(N)";
        OEGraphMol refmol = new OEGraphMol();
        oechem.OESmilesToMol(refmol, refsmiles);
        oedepict.OEPrepareDepiction(refmol);
        OESubSearch ss = new OESubSearch(refmol,
                OEExprOpts.DefaultAtoms,
                OEExprOpts.DefaultBonds);

        ArrayList<String> aminosmiles = new ArrayList<String>();
        aminosmiles.add("O=C(O)[C@@H](N)Cc1c[nH]cn1 Histidine");
        aminosmiles.add("CC(C)[C@@H](C(=O)O)N Valine");
        aminosmiles.add("C[C@H]([C@@H](C(=O)O)N)O Threonine");
        aminosmiles.add("C1C[C@H](NC1)C(=O)O Proline");

        ArrayList<OEGraphMol> aminoacids = new ArrayList<OEGraphMol>();
        for (String smiles : aminosmiles) {
            OEGraphMol mol = new OEGraphMol();
            oechem.OESmilesToMol(mol, smiles);
            oedepict.OEPrepareDepiction(mol);
            aminoacids.add(mol);
        }

        OEImage image = new OEImage(400, 400);

        int rows = 2;
        int cols = 2;
        OEImageGrid grid = new OEImageGrid(image, rows, cols);

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(grid.GetCellWidth(),
                grid.GetCellHeight(),
                OEScale.AutoScale);
        opts.SetAtomStereoStyle(OEAtomStereoStyle.Display.All);

        double minscale = Double.MAX_VALUE;
        for (OEGraphMol mol : aminoacids) {
            minscale = Math.min(minscale, oedepict.OEGetMoleculeScale(mol, opts));
        }
        opts.SetScale(minscale);

        int hstyle = OEHighlightStyle.Stick;
```

(continues on next page)

(continued from previous page)

```
        OEImageBaseIter celliter = grid.GetCells();
        for (int a = 0; a < aminoacids.size() && celliter.IsValid(); ++a, celliter.
↪Increment()) {

            OEGraphMol mol = aminoacids.get(a);
            OEAlignmentResult alignres = new OEAlignmentResult(oedepict.
↪OEPrepareAlignedDepiction(mol, ss));
            OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
            if (alignres.IsValid()) {
                oedepict.OEAddHighlighting(disp, oechem.getOELightGreen(), hstyle,␣
↪alignres);
            }
            oedepict.OERenderMolecule(celliter.Target(), disp);
        }

        oedepict.OEWriteImage("SubSearchAlign.png", image);
    }
}
```

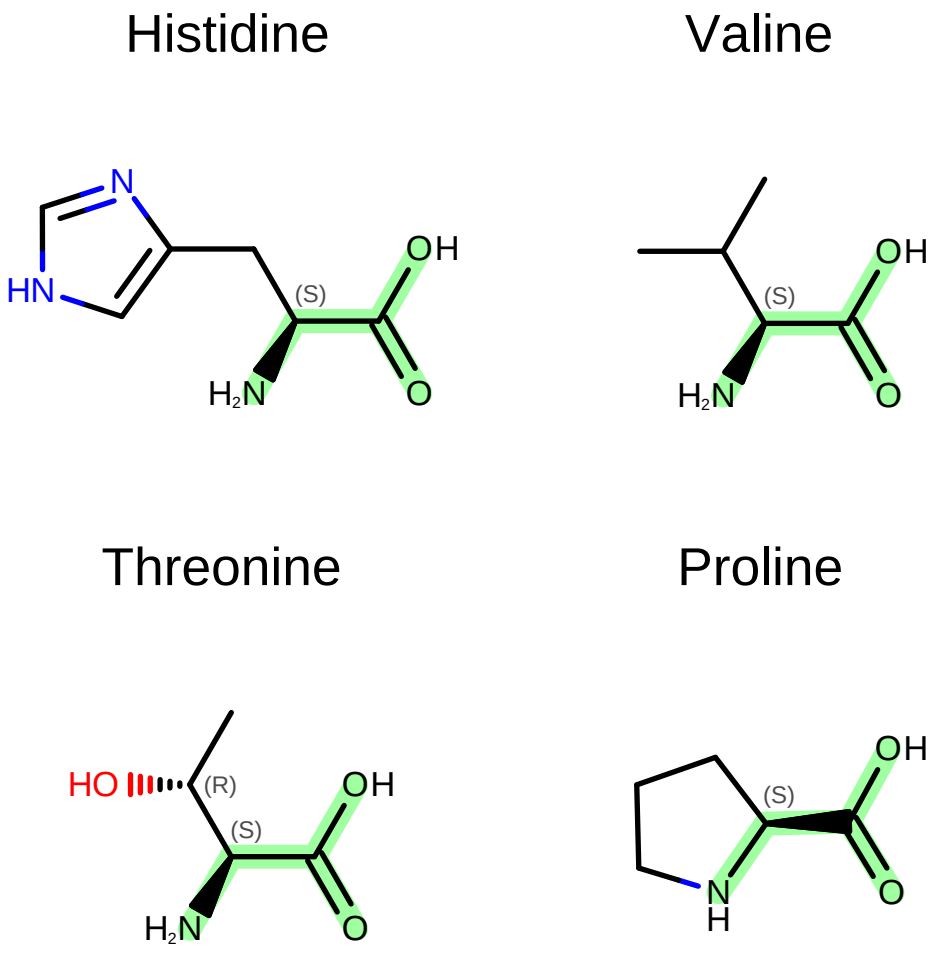


Fig. 34: **Example of depiction with alignment based on substructure match**

See also

- OESubSearch class in the **OEChem TK** manual
- *OEImageGrid* class
- `OEAddHighlighting` function
- `OERenderMolecule` function
- `OEWriteImage` function
- *Aligning Molecules Based on Substructure Search* example

2.8.3 Molecule Alignment Based on Molecular Similarity

The *Figure: Example of depiction without alignment* is generated by keeping the original orientation of the compounds.

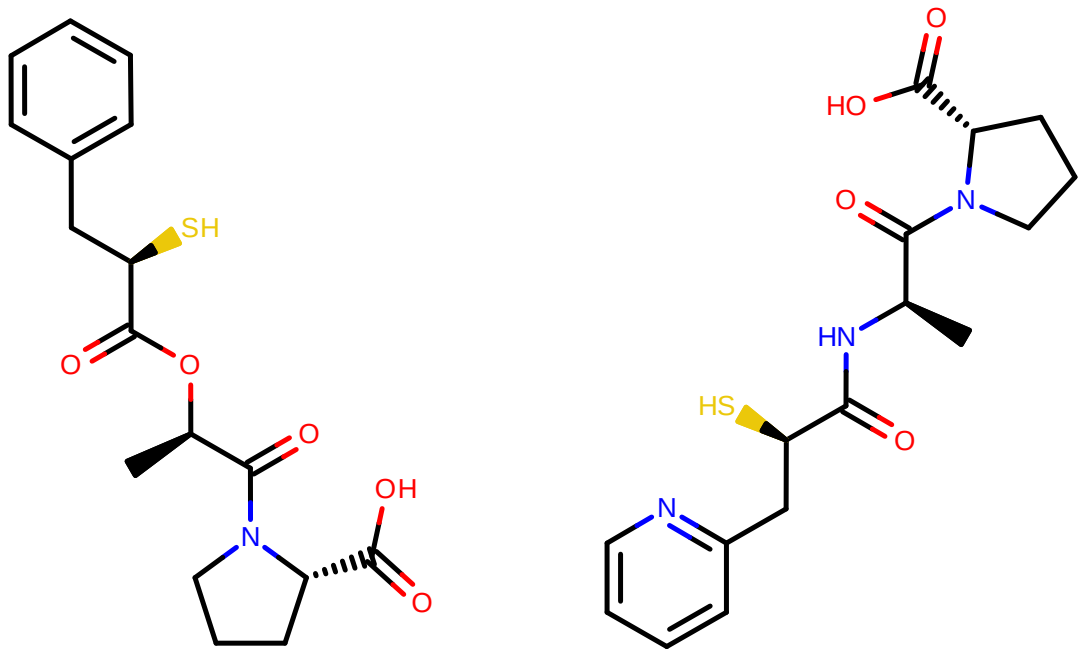


Fig. 35: **Example of depiction without alignment**

The `Listing 3` code example shows how to align these molecules based on their molecular similarity.

- First, two molecules are initialized from SMILES string and prepared for depiction.
- An *OEImageGrid* object is constructed that allows to render the two molecules in a grid next to each other.
- The *fit* molecule is then aligned to the *reference*. The `OEGetFPOverlap` function is utilized to return all common fragments found between two molecules based on a given fingerprint type. These common fragments reveal the similar parts of the two molecules being compared that are used by the `OEPrepareMultiAlignedDepiction` function to find the best alignment between the molecules. See more fingerprint types in the Fingerprint Types chapter of the **GraphSim TK** manual.

- An *OE2DMolDisplayOptions* object is initialized that stores the properties determine how the molecules are rendered. In order to render the molecules in equal size the smallest depiction scaling factor is calculated.
- Then both molecules are rendered into the separate cells of the grid layout, the image is written into a `png` file.

The image created by *Listing 3* is shown in *Figure: Example of depiction with alignment based on molecular similarity*.

> **Note**
>
> **GraphSim TK** license in **not** required to run the *Listing 3* example.

Listing 3: Example of molecule alignment based on molecular similarity

```
public class FPAlign {
    public static void main(String argv[]) {
        OEGraphMol refmol = new OEGraphMol();
        oechem.OESmilesToMol(refmol,
↪"C[C@H](C(=O)N1CCC[C@H]1C(=O)O)OC(=O)[C@@H](Cc2ccccc2)S");
        oedepict.OEPrepareDepiction(refmol);

        OEGraphMol fitmol = new OEGraphMol();
        oechem.OESmilesToMol(fitmol,
↪"C[C@H](C(=O)N1CCC[C@H]1C(=O)O)NC(=O)[C@@H](Cc2ccccn2)S");
        oedepict.OEPrepareDepiction(fitmol);

        OEImage image = new OEImage(500, 300);

        int rows = 1;
        int cols = 2;
        OEImageGrid grid = new OEImageGrid(image, rows, cols);

        OEFPTypeBase fptype = oegraphsim.OEGetFPType(OEFPType.Tree);
        oedepict.OEPrepareMultiAlignedDepiction(fitmol, refmol,
              oegraphsim.OEGetFPOverlap(refmol, fitmol, fptype));

        OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(grid.GetCellWidth(),
                grid.GetCellHeight(),
                OEScale.AutoScale);
        opts.SetTitleLocation(OETitleLocation.Hidden);

        double refscale = oedepict.OEGetMoleculeScale(refmol, opts);
        double fitscale = oedepict.OEGetMoleculeScale(fitmol, opts);
        opts.SetScale(Math.min(refscale, fitscale));

        OEImageBase refcell = grid.GetCell(1, 1);
        OE2DMolDisplay refdisp = new OE2DMolDisplay(refmol, opts);
        oedepict.OERenderMolecule(refcell, refdisp);

        OEImageBase fitcell = grid.GetCell(1, 2);
        OE2DMolDisplay fitdisp = new OE2DMolDisplay(fitmol, opts);
        oedepict.OERenderMolecule(fitcell, fitdisp);
```

(continues on next page)

(continued from previous page)

```
        oedepict.OEWriteImage("FPAlign.png", image);
    }
}
```

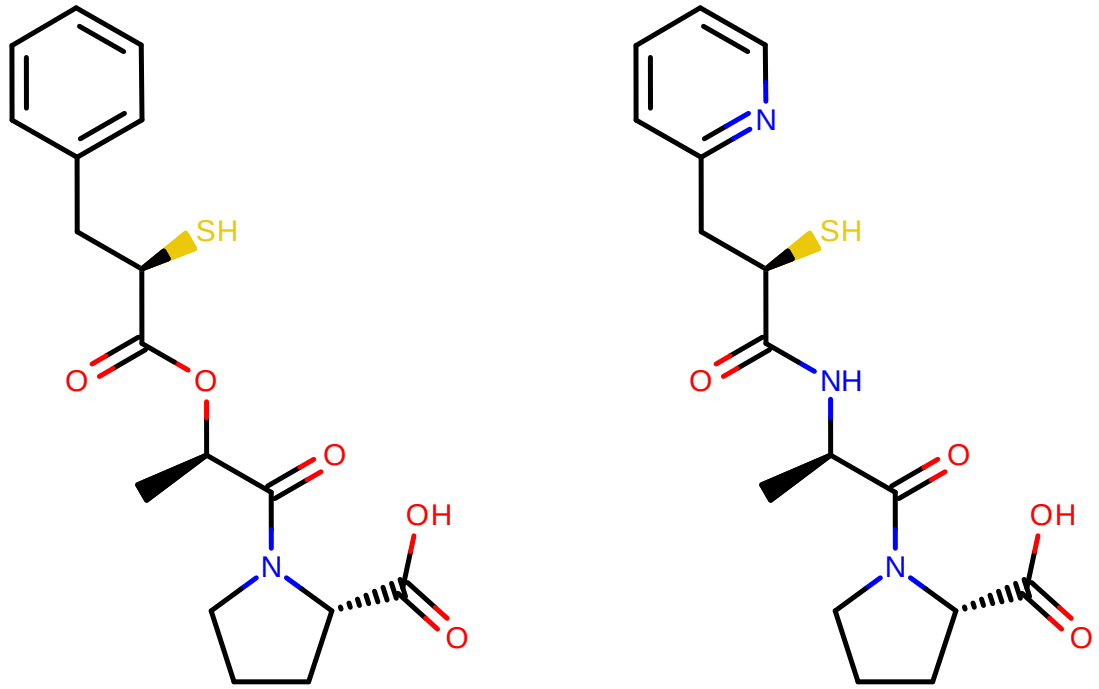
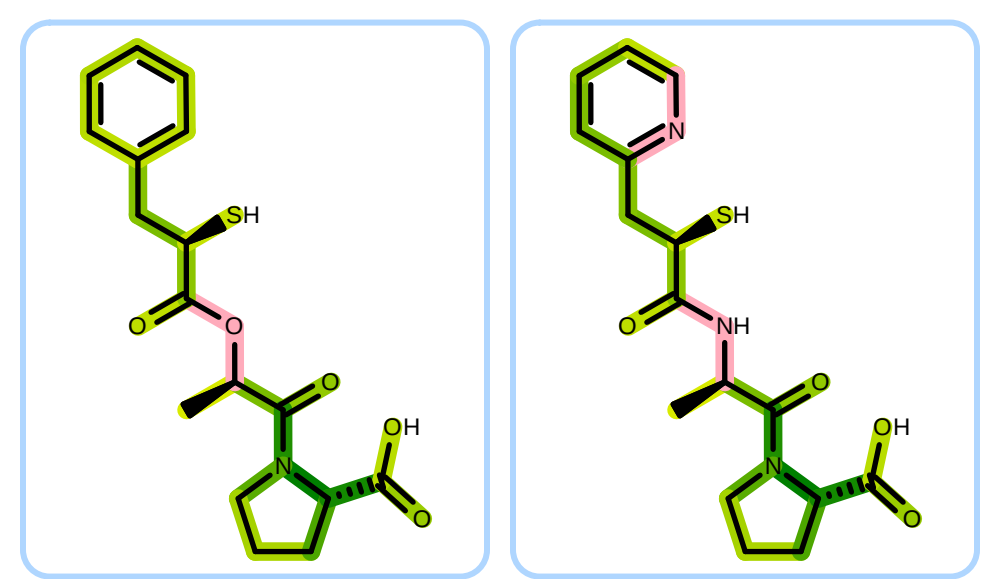


Fig. 36: **Example of depiction with alignment based on molecule similarity**

See also

- *OEImageGrid* class
- *OERenderMolecule* function
- *OEWriteImage* function

See also



The Python script that visualizes molecule similarity based on fingerprints can be downloaded from the OpenEye Python Cookbook

2.9 Generating Interactive SVG Images

2.9.1 SVG group

SVG group is a container for grouping and naming collections of SVG drawing elements. In **OEDepict TK**, the *OESVGGroup* class is to used for grouping drawing instructions together.

The code snippet below shows how to create and use an *OESVGGroup* object.

```
OESVGGroup svg_group_circ = image.NewSVGGroup("circle");

image.PushGroup(svg_group_circ);
image.DrawCircle(oedepict.OEGetCenter(image), 30, oedepict.getOERedBoxPen());
image.PopGroup(svg_group_circ);

oedepict.OEWriteImage("CreateSVGGroup.svg", image);
```

- First an *OESVGGroup* object is created on the image by calling the *OEImageBase.NewSVGGroup* method.
- The *OESVGGroup* object is the can be utilized by invoking the *OEImageBase.PushGroup* method. This result in writing the `<g id='circle'>` line into the `svg` image.
- Each push has to be coupled with a corresponding pop. When the *OEImageBase.PopGroup* method is invoked the `</g>` closing line will be written into the image file.

The above snippet will generate the following `svg` image:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
```

(continues on next page)

(continued from previous page)

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
↪viewBox="0.00 0.00 100.00 100.00">
<!-- Created by OpenEye Scientific Software -->
<!-- Creation Date Fri Mar  3 16:37:06 2017 -->
<rect width="100.00" height="100.00" fill="white" />
<g id='circle'>
<circle cx="50.00" cy="50.00" r="30.00" fill="red" stroke="red" stroke-linejoin="round"
↪stroke-linecap="round" />
</g>
</svg>
```

The **guideline** below should be followed when using SVG groups in **OEDepict TK**:

- The name of the *OESVGGroup* has to be unique for the given image and it can only contain alphanumeric, _ or - characters.
- An *OESVGGroup* object can be used (pushed / popped) only once.
- *OESVGGroup* objects can be nested. They should be used in LIFO (for last in, first out) i.e. the group that has been pushed last should be popped first.

See also

- Group SVG element section in MDN
- Grouping section in W3C

2.9.2 SVG class

In **OEDepict TK**, the *OESVGClass* class can be used to add class attributes to group containers.

The code snippet below shows how to create and use an *OESVGClass* object.

```
OESVGGroup svg_group_circ = image.NewSVGGroup("circle");
OESVGGroup svg_group_rect = image.NewSVGGroup("rectangle");
OESVGClass svg_class_objs = image.NewSVGClass("object");

svg_group_circ.AddSVGClass(svg_class_objs);
svg_group_rect.AddSVGClass(svg_class_objs);

image.PushGroup(svg_group_circ);
image.DrawCircle(oedepict.OEGetCenter(image), 30, oedepict.getOERedBoxPen());
image.PopGroup(svg_group_circ);

image.PushGroup(svg_group_rect);
image.DrawRectangle(new OE2DPoint(10, 10), new OE2DPoint(90, 90),
                    oedepict.getOELightGreyPen());
image.PopGroup(svg_group_rect);

oedepict.OEWriteImage("CreateSVGClass.svg", image);
```

- First two *OESVGGroup* objects are created on the image, followed by creating a *OESVGClass* object by calling the *OEImageBase.NewSVGClass* method.
- The *OESVGClass* object then can be added to any groups by calling *OESVGGroup.AddSVGClass* method.

- When a *OESVGGroup* object is pushed classes associated with it will be written into the `svg` file: `<g id='circle' class='object'>` and `<g id='rectangle' class='object'>`.

The above snippet will generate the following `svg` image:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"  ␣
↪viewBox="0.00 0.00 100.00 100.00">
<!-- Created by OpenEye Scientific Software -->
<!-- Creation Date Fri Mar  3 17:22:19 2017 -->
<rect width="100.00" height="100.00" fill="white" />
<g id='circle' class='object'>
<circle cx="50.00" cy="50.00" r="30.00" fill="red" stroke="red" stroke-linejoin="round"␣
↪stroke-linecap="round" />
</g>
<g id='rectangle' class='object'>
<rect x="10" y="10" width="80" height="80" fill="none" stroke="#bfbfbf" stroke-linejoin=
↪"round" stroke-linecap="round" />
</g>
</svg>
```

The **guideline** below should be followed when using SVG classes in **OEDepict TK**:

- The name of the *OESVGClass* has to be unique for the given image and it can only contain alphanumeric, _ or - characters.
- An *OESVGClass* object can be associated with multiple *OESVGGroup* objects.
- An *OESVGGroup* object can be associated with multiple *OESVGClass* objects.

See also

- MDN Class attribute section in MDN
- WSC Class attribute section in WSC

2.9.3 Examples of Generating Interactive SVG Images

OEDepict TK also provides high-level APIs that are not only group SVG elements together but also adds styles and events to them to generate interactive images.

See the following **OEDepict TK** API:

- *`OEAddSVGHover`* and *`OEAddSVGToggle`* low-level functions that allow to add effects to any objects.
- *`OEAddSVGClickEvent`* low-level functions that allows to add click event to any objects.
- *`OEDrawSVGHoverText`* function that displays a text associated with an atom or bond when the mouse is hovering over their position on the image.
- *`OEDrawSVGToggleText`* function that displays a text associated with an atom or bond when clicking at their position on the image.

See also the following **OEGrapheme TK** APIs that generate interactive SVG images:

- `OERenderContactMap` function that generated an interactive protein-ligand contact map image. Clicking on any residue will highlight those ligand atoms that are interacting with the residue.

- `OEDrawIridiumData` function
- `OEDrawPeptide` function
- `OERenderRamachandranPlot` function
- `OERenderBFactorMap` function

2.9.4 Inserting SVG Images into HTML

The interactive `svg` image generated by **OEDepict TK** and **Grapheme TK** can inserted into HTML files with the SVG MIME type:

```
<object data="<imagename>.svg" type="image/svg+xml"></object>
```

The following example inserts two `svg` images into an HTML table:

```
<!DOCTYPE html>
<html>
<head>
  <title>test</title>
</head>
<body>

<h2>Examples of interactive SVG images inserted into HTML</h2>

<table style="width:100%">
  <tr>
    <th>hover atom text</th>
    <th>toggle atom text</th>
  </tr>
  <tr>
   <td><object data="hover-atom-text.svg" type="image/svg+xml"  width = "400" ></object>
↪</td>
   <td><object data="toggle-atom-text.svg" type="image/svg+xml"  width = "400" ></object>
↪</td>
  </tr>

</table>
</body>
</html>
```

Download the following three files and open the `insert-index.html` file in a browser.

2.9.5 How to Catch SVG Events

The example in this section illustrates how to throw a mouse event in an `.svg` image and how to catch it in a `html` file.

The code below shows how to create an `.svg` image where each atom is associated with an SVG event. When clicking on an atom in the image, a click event is fired with the message specified in the *OEAddSVGClickEvent* function.

The events thrown by the `AddAtomClickEvent.svg` image can be caught in the `html` file that contains the image. The following `html` file illustrates how to catch the events and display the messages that were associated with the atoms using the *OEAddSVGClickEvent* function.

```
<!DOCTYPE html>
<html>
```

(continues on next page)

(continued from previous page)

```
<head>
  <title>test</title>

  <script type="text/javascript">
    window.addEventListener("load", init);

    function init() {
      var svg = document.getElementById('depictsvg');
      svg.contentDocument.addEventListener('OEDepictSVGEvent', function(evt) {
        var logarea = document.querySelector('.svg-event-log');
        logarea.innerHTML = evt.detail.message;
      });
    }
  </script>

</head>
<body>

<h2> Image </h2>
<object data="AddAtomClickEvent.svg" type="image/svg+xml" id="depictsvg" width = "400" >
↪</object>

<h2>Event log message:</h2>
<h4 class="svg-event-log">

</h4>

</body>
</html>
```

To try it out follow the instructions below:

1. Generate a directory (such as `test`) that contains both the `AddAtomClickEvent.svg` and the `index.html` file (click on them to download).
2. Start up a HTTP server inside `test` directory

   ```
   prompt> python -m http.server
   ```

3. Open http://0.0.0.0:8000 in your browser (see image below)
4. Click on any atom on the image and the message specified by the *OEAddSVGClickEvent* function will be displayed below the image.

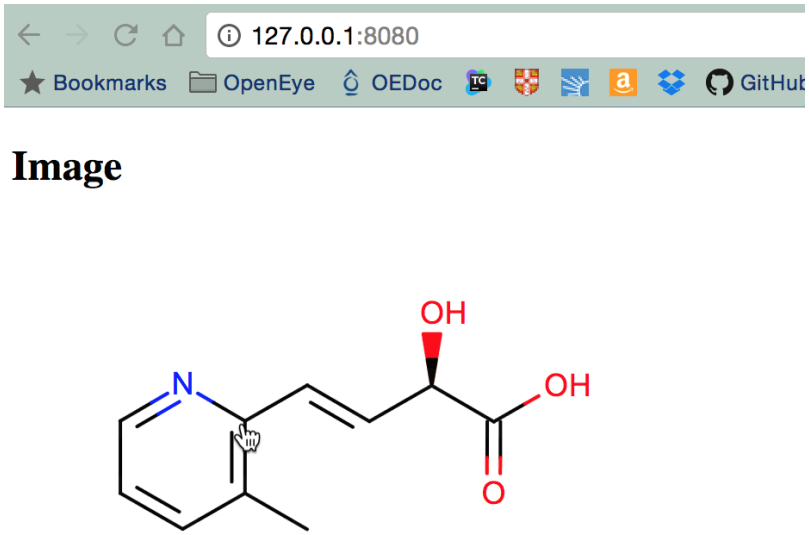


Fig. 37: **Screenshot of the OEAddSVGClickEvent example in the browser**

CHAPTER

THREE

EXAMPLES

3.1 OEDepict Examples

The following table lists the currently available **OEDepict TK** examples:

Program	Description
Depict	generating 2D coordinates
Match2Img	depicting substructure search match
MCSAlign2D	aligning molecules based on MCS
MDLQuery2Img	depicting MDL query
MDLReaction2Img	depicting MDL reaction
Mol2Img	depicting a single molecule
Mols2PDF	depicting molecules in a multi-page image
Mols2Img	depicting molecules in a grid
RingDict2PDF	generating 2D ring dictionary report
SMARTSAlign2D	aligning molecules based on substructure search
ViewMDLSearch	depicting MDL query substructure search hits

Examples:

3.1.1 Generating 2D coordinates

A program that assigns 2D coordinates to molecule(s).

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> java Depict --help
```

will generate the following output:

```
Simple parameter list
input/output options :
   -in : Input filename
   -out : Output filename
   -ringdict : User-defined 2D ring dictionary

 prepare depiction options :
   -orientation : Set the preferred orientation of 2D coordinates
   -suppressH : Suppress explicit hydrogens of molecule(s)
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
/*****************************************************************************
 * Generates 2D coordinates
 *****************************************************************************/
package openeye.examples.oedepict;

import java.net.URL;
import java.util.*;

import openeye.oechem.*;
import openeye.oedepict.*;

public class Depict {

    public static void main(String argv[]) {
        URL fileURL = Depict.class.getResource("Depict.txt");
        try {
            OEInterface itf = new OEInterface();
            oechem.OEConfigureFromURL(itf, fileURL);
            oedepict.OEConfigurePrepareDepictionOptions(itf, OEPrepareDepictionSetup.
↪SuppressHydrogens |
                                                        OEPrepareDepictionSetup.
↪DepictOrientation);

            if (!oechem.OEParseCommandLine(itf, argv, "Depict"))
                oechem.OEThrow.Fatal("Unable to interpret command line!");

            oemolistream ifs = new oemolistream();
            if (!ifs.open(itf.GetString("-in")))
                oechem.OEThrow.Fatal("Cannot open input file!");

            oemolostream ofs = new oemolostream(".sdf");
            if (itf.HasString("-out")) {
                String oname = itf.GetString("-out");
```

(continues on next page)

(continued from previous page)

```
                if (!ofs.open(oname))
                    oechem.OEThrow.Fatal("Cannot open output file!");
                if (!oechem.OEIs2DFormat(ofs.GetFormat()))
                    oechem.OEThrow.Fatal("Invalid output format for 2D coordinates");
            }

            if (itf.HasString("-ringdict")) {
                String rdfname = itf.GetString("-ringdict");
                if (!oechem.OEInit2DRingDictionary(rdfname))
                    oechem.OEThrow.Warning("Cannot use user-defined ring dictionary!");
            }

            OEPrepareDepictionOptions popts = new OEPrepareDepictionOptions();
            oedepict.OESetupPrepareDepictionOptions(popts, itf);
            popts.SetClearCoords(true);

            OEGraphMol mol = new OEGraphMol();
            while (oechem.OEReadMolecule(ifs, mol)) {
                oedepict.OEPrepareDepiction(mol, popts);
                oechem.OEWriteMolecule(ofs, mol);
            }
            ifs.close();
            ofs.close();
        }
        catch (java.io.IOException e) {
            System.err.println("Unable to open interface file");
        }
    }
}
```

with interface file

```
# Depict interface file
!BRIEF [-in] <input> [-out] <output> [-ringdict] <rd file>

!CATEGORY "input/output options :"

  !PARAMETER -in
    !ALIAS -i
    !TYPE string
    !REQUIRED true
    !KEYLESS 1
    !VISIBILITY simple
    !BRIEF Input filename
  !END

  !PARAMETER -out
    !ALIAS -o
    !TYPE string
    !REQUIRED false
    !KEYLESS 2
    !VISIBILITY simple
```

(continues on next page)

(continued from previous page)

```
    !BRIEF Output filename
  !END

  !PARAMETER -ringdict
    !ALIAS -rd
    !TYPE string
    !REQUIRED false
    !VISIBILITY simple
    !BRIEF User-defined 2D ring dictionary
    !DETAIL
        2D ring dictionaries can be generated by the following OEChem examples:
        C++    - createringdict.cpp
        Python - createringdict.py
        Java   - CreateRingDict.java
        C#     - CreateRingDict.cs
  !END

!END
```

See also

- *OEConfigurePrepareDepictionOptions* function
- OEIs2DFormat function
- *OEPrepareDepictionOptions* class
- *OESetupPrepareDepictionOptions* function
- *OEPrepareDepiction* function

Example

```
prompt> java Depict .ism .mol

C(=O)N
```

will generate the following output:

```
  -OEChem-07031123462D

  3  2  0     0  0  0  0  0  0999 V2000
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.0000    0.0000    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -0.5000   -0.8660    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  2  0  0  0  0
  1  3  1  0  0  0  0
M  END
```

Note

The above program can generate 2D coordinates with user-defined 2D ring layouts when using the **-ringdict** pa-

rameter.

See also the 2D Coordinate Generation Examples section of the **OEChem TK** manual that shows examples about how to generate and utilize user-defined ring dictionaries.

3.1.2 Depicting Substructure Search Match

A program that performs a substructure search initialized with the given SMARTS pattern and highlights the identified matches on the depicted molecule.

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> java Match2Img --help
```

will generate the following output:

```
Simple parameter list
 highlight options :
   -highlightcolor : Highlighting color
   -highlightstyle : Highlighting style

 image options :
   -height : Height of output image
   -width : Width of output image

 input/output options :
   -in : Input filename
   -out : Output filename
   -smarts : SMARTS pattern

 molecule display options :
   -aromstyle : Aromatic ring display style
   -atomcolor : Atom coloring style
   -atomlabelfontscale : Atom label font scale
   -atomprop : Atom property display
   -atomstereostyle : Atom stereo display style
   -bondcolor : Bond coloring style
   -bondprop : Bond property display
   -bondstereostyle : Bond stereo display style
   -hydrstyle : Hydrogen display style
   -linewidth : Default bond line width
   -protgroupdisp : Protective group display style
   -scale : Scaling of the depicted molecule
   -superdisp : Super atom display style
   -titleloc : Location of the molecule title

 prepare depiction options :
   -clearcoords : Clear and regenerate 2D coordinates of molecule(s)
   -orientation : Set the preferred orientation of 2D coordinates
   -suppressH : Suppress explicit hydrogens of molecule(s)
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
/*****************************************************************************
 * Depict a molecule and highlight the substructure specified by
 * the given SMARTS pattern
 *****************************************************************************/
package openeye.examples.oedepict;

import java.net.URL;
import java.util.*;

import openeye.oechem.*;
import openeye.oedepict.*;

public class Match2Img {

    public static void main(String argv[]) {
        URL fileURL = Match2Img.class.getResource("Match2Img.txt");
        try {
            OEInterface itf = new OEInterface();
            oechem.OEConfigureFromURL(itf, fileURL);
            oedepict.OEConfigureImageOptions(itf);
            oedepict.OEConfigurePrepareDepictionOptions(itf);
            oedepict.OEConfigure2DMolDisplayOptions(itf);
            oedepict.OEConfigureHighlightParams(itf);

            if (!oechem.OEParseCommandLine(itf, argv, "Match2Img")) {
                oechem.OEThrow.Fatal("Unable to interpret command line!");
            }

            String iname = itf.GetString("-in");
            String oname = itf.GetString("-out");

            String ext = oechem.OEGetFileExtension(oname);
            if (!oedepict.OEIsRegisteredImageFile(ext)) {
```

(continues on next page)

(continued from previous page)

```
            oechem.OEThrow.Fatal("Unknown image type!");
        }

        oemolistream ifs = new oemolistream();
        if (!ifs.open(iname)) {
            oechem.OEThrow.Fatal("Cannot open input file!");
        }

        oeofstream ofs = new oeofstream();
        if (!ofs.open(oname)) {
            oechem.OEThrow.Fatal("Cannot open output file!");
        }

        OEGraphMol mol = new OEGraphMol();
        if (!oechem.OEReadMolecule(ifs, mol)) {
            oechem.OEThrow.Fatal("Cannot read input file!");
        }
        ifs.close();

        String  smarts = itf.GetString("-smarts");

        OESubSearch ss = new OESubSearch();
        if (!ss.Init(smarts)) {
            oechem.OEThrow.Fatal("Cannot parse smarts: " + smarts);
        }

        OEPrepareDepictionOptions popts = new OEPrepareDepictionOptions();
        oedepict.OESetupPrepareDepictionOptions(popts, itf);
        oedepict.OEPrepareDepiction(mol, popts);

        double width  = oedepict.OEGetImageWidth (itf);
        double height = oedepict.OEGetImageHeight(itf);
        OE2DMolDisplayOptions dopts = new OE2DMolDisplayOptions(width, height,␣
↪OEScale.AutoScale);
        oedepict.OESetup2DMolDisplayOptions(dopts, itf);
        dopts.SetMargins(10.0);

        OE2DMolDisplay disp = new OE2DMolDisplay(mol, dopts);

        int     hstyle = oedepict.OEGetHighlightStyle(itf);
        OEColor hcolor = oedepict.OEGetHighlightColor(itf);

        oechem.OEPrepareSearch(mol, ss);

        boolean unique = true;
        for (OEMatchBase match : ss.Match(mol, unique)) {
            oedepict.OEAddHighlighting(disp, hcolor, hstyle, match);
        }

        oedepict.OERenderMolecule(ofs, ext, disp);
        ofs.close();
    }
```

(continues on next page)

(continued from previous page)

```
        catch (java.io.IOException e) {
            System.err.println("Unable to open interface file");
        }
    }
}
```

with interface file

```
# Match2Img interface file
!BRIEF [-in] <input> [-smarts] <smarts> [-out] <output image>

!CATEGORY "input/output options :"

    !PARAMETER -in
      !ALIAS -i
      !TYPE string
      !REQUIRED true
      !KEYLESS 1
      !VISIBILITY simple
      !BRIEF Input filename
    !END

    !PARAMETER -smarts
      !TYPE string
      !REQUIRED true
       !KEYLESS 2
      !VISIBILITY simple
      !BRIEF SMARTS pattern
    !END

    !PARAMETER -out
      !ALIAS -o
      !TYPE string
      !REQUIRED true
      !KEYLESS 3
      !VISIBILITY simple
      !BRIEF Output filename
    !END

!END
```

See also

- *OEConfigureImageOptions* function
- *OEConfigurePrepareDepictionOptions* function
- *OEConfigure2DMolDisplayOptions* function
- *OEConfigureHighlightParams* function
- *OESetup2DMolDisplayOptions* function
- OESubSearch class

- *OEPrepareDepictionOptions* class
- *OESetupPrepareDepictionOptions* function
- *OEPrepareDepiction* function
- *OEGetImageWidth* and *OEGetImageHeight* functions
- *OE2DMolDisplayOptions* class
- *OESetup2DMolDisplayOptions* function
- *OE2DMolDisplay* class
- *OEGetHighlightStyle* and *OEGetHighlightColor* functions
- OEPrepareSearch function
- *OEAddHighlighting* function
- *OERenderMolecule* function

Examples

```
prompt> java Match2Img -in indole.ism -out image.png -smarts "c1ccccc1"
```

will generate the image shown in *Figure: Example of using the program with default highlighting*

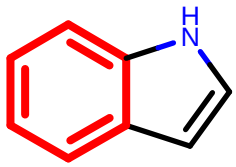


Fig. 1: **Example of using the program with default highlighting**

```
prompt> java Match2Img -in indole.ism -out image.png -smarts "c1ccccc1" -highlightcolor␣
↪green
```

will generate the image shown in *Figure: Example of using the program with green highlighting*

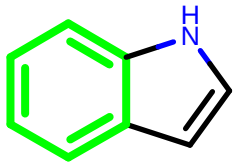


Fig. 2: **Example of using the program with green highlighting**

```
prompt> java Match2Img -in indole.ism -out image.png -smarts "c1ccccc1" -highlightstyle␣
↪BallAndStick
```

will generate the image shown in *Figure: Example of using the program with ball and stick*

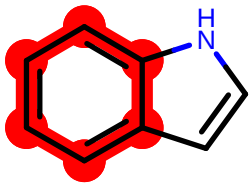


Fig. 3: **Example of using the program with ball and stick highlighting**

See also

- *Highlighting* chapter

3.1.3 Aligning Molecule Based on MCS

A program that aligns molecules based on their maximum common substructure.

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> java MCSAlign2D --help
```

will generate the following output:

```
Simple parameter list
 input/output options :
   -fit : Align filename
   -out : Output filename
   -ref : Ref filename
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
```

(continues on next page)

(continued from previous page)

```
liable for any damages or liability in connection with the Sample Code
or its use.
*/
/*****************************************************************************
 * Aligns the fit molecules based on their maximum common substructure
 * with the reference molecule.
 *****************************************************************************/
package openeye.examples.oedepict;

import java.net.URL;
import java.util.*;

import openeye.oechem.*;
import openeye.oedepict.*;

public class MCSAlign2D {

    public static void main(String argv[]) {
        URL fileURL = MCSAlign2D.class.getResource("MCSAlign2D.txt");
        try {
            OEInterface itf = new OEInterface();
            oechem.OEConfigureFromURL(itf, fileURL);

            if (!oechem.OEParseCommandLine(itf, argv, "MCSAlign2D"))
                oechem.OEThrow.Fatal("Unable to interpret command line!");

            String rname = itf.GetString("-ref");
            String fname = itf.GetString("-fit");
            String oname = itf.GetString("-out");

            oemolistream rifs = new oemolistream();
            if (!rifs.open(rname))
                oechem.OEThrow.Fatal("Cannot open reference molecule file!");

            OEGraphMol refmol = new OEGraphMol();
            if (!oechem.OEReadMolecule(rifs, refmol))
                oechem.OEThrow.Fatal("Cannot read reference molecule!");
            rifs.close();

            oemolistream fifs = new oemolistream();
            if (!fifs.open(fname))
                oechem.OEThrow.Fatal("Cannot open align molecule file!");

            oemolostream ofs = new oemolostream();
            if (!ofs.open(oname))
                oechem.OEThrow.Fatal("Cannot open output file!");
            if (!oechem.OEIs2DFormat(ofs.GetFormat()))
                oechem.OEThrow.Fatal("Invalid output format for 2D coordinates");

            oedepict.OEPrepareDepiction(refmol);

            OEMCSSearch mcss = new OEMCSSearch(OEMCSType.Approximate);
```

(continues on next page)

(continued from previous page)

```
            int atomexpr = OEExprOpts.DefaultAtoms;
            int bondexpr = OEExprOpts.DefaultBonds;
            mcss.Init(refmol, atomexpr, bondexpr);
            mcss.SetMCSFunc(new OEMCSMaxBondsCompleteCycles());

            oechem.OEWriteConstMolecule(ofs, refmol);

            OEGraphMol fitmol = new OEGraphMol();
            while (oechem.OEReadMolecule(fifs, fitmol)) {
                OEAlignmentResult alignres = new OEAlignmentResult(oedepict.
↪OEPrepareAlignedDepiction(fitmol, mcss));
                if (alignres.IsValid()) {
                    oechem.OEThrow.Info(fitmol.GetTitle() + "  mcs size: " + alignres.
↪NumAtoms());
                    oechem.OEWriteMolecule(ofs, fitmol);
                }
            }
            fifs.close();
            ofs.close();
        }
        catch (java.io.IOException e) {
            System.err.println("Unable to open interface file");
        }
    }
}
```

with interface file

```
# MCSAlign2D interface file
!BRIEF [-ref] <input> [-fit] <input> [-out] <output>

!CATEGORY "input/output options :"

    !PARAMETER -ref
      !ALIAS -r
      !TYPE string
      !REQUIRED true
      !KEYLESS 1
      !VISIBILITY simple
      !BRIEF Ref filename
    !END

    !PARAMETER -fit
      !ALIAS -f
      !TYPE string
      !REQUIRED true
      !KEYLESS 2
      !VISIBILITY simple
      !BRIEF Align filename
    !END

    !PARAMETER -out
```

(continues on next page)

(continued from previous page)

```
      !ALIAS -o
      !TYPE string
      !REQUIRED true
      !KEYLESS 3
      !VISIBILITY simple
      !BRIEF Output filename
    !END

!END
```

See also

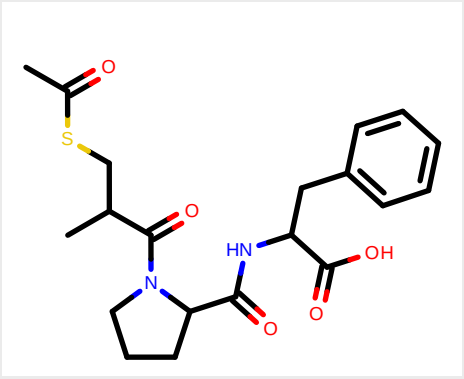
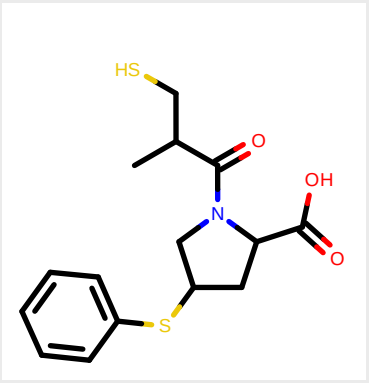
- *Molecule Alignment* chapter
- `OEIs2DFormat` function
- *`OEPrepareDepiction`* function
- OEMCSSearch class
- *`OEPrepareAlignedDepiction`* function

Example

```
prompt> java MCSAlign2D -ref ace1.ism -fit ace2.ism -out out.mol
```

taking the output coordinates and generating the images with the **Mol2Img** program will produce the images shown in *Table: Example of using the program to align two molecules based on MCS*.

Table 1: Example of using the program to align two molecules based on MCS

reference molecule	aligned molecule
	

See also

- *Molecule Alignment* chapter

3.1.4 Depicting MDL Query

A program that converts an MDL query structure into an image file.

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> java MDLQuery2Img --help
```

will generate the following output:

```
Simple parameter list
 image options :
   -height : Height of output image
   -width : Width of output image

 input/output options :
   -in : Input filename
   -out : Output filename

 molecule display options :
   -aromstyle : Aromatic ring display style
   -atomcolor : Atom coloring style
   -atomlabelfontscale : Atom label font scale
   -atomprop : Atom property display
   -atomstereostyle : Atom stereo display style
   -bondcolor : Bond coloring style
   -bondprop : Bond property display
   -bondstereostyle : Bond stereo display style
   -hydrstyle : Hydrogen display style
   -linewidth : Default bond line width
   -protgroupdisp : Protective group display style
   -scale : Scaling of the depicted molecule
   -superdisp : Super atom display style
   -titleloc : Location of the molecule title
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
```

(continues on next page)

(continued from previous page)

```
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
/*****************************************************************************
 * Converts an MDL query structure into an image file.
 * The output file format depends on its file extension.
 *****************************************************************************/
package openeye.examples.oedepict;

import java.net.URL;
import java.util.*;

import openeye.oechem.*;
import openeye.oedepict.*;

public class MDLQuery2Img {

    public static void main(String argv[]) {
        URL fileURL = MDLQuery2Img.class.getResource("MDLQuery2Img.txt");
        try {
            OEInterface itf = new OEInterface();
            oechem.OEConfigureFromURL(itf, fileURL);
            oedepict.OEConfigureImageOptions(itf);
            oedepict.OEConfigure2DMolDisplayOptions(itf);

            if (!oechem.OEParseCommandLine(itf, argv, "MDLQuery2Img"))
                oechem.OEThrow.Fatal("Unable to interpret command line!");

            String iname = itf.GetString("-in");
            String oname = itf.GetString("-out");

            String ext = oechem.OEGetFileExtension(oname);
            if (!oedepict.OEIsRegisteredImageFile(ext))
                oechem.OEThrow.Fatal("Unknown image type!");

            oemolistream ifs = new oemolistream();
            if (!ifs.open(iname))
                oechem.OEThrow.Fatal("Cannot open input file!");

            if (ifs.GetFormat() != OEFormat.MDL)
                oechem.OEThrow.Fatal("Input file is not an MDL query file");

            oeofstream ofs = new oeofstream();
            if (!ofs.open(oname))
                oechem.OEThrow.Fatal("Cannot open output file!");
```

(continues on next page)

(continued from previous page)

```
            OEGraphMol mol = new OEGraphMol();
            if (!oechem.OEReadMDLQueryFile(ifs, mol))
                oechem.OEThrow.Fatal("Cannot read mdl query input file!");
            ifs.close();

            boolean clearcoords = false;
            boolean suppressH   = false;
            oedepict.OEPrepareDepiction(mol, clearcoords, suppressH);

            double width  = oedepict.OEGetImageWidth (itf);
            double height = oedepict.OEGetImageHeight(itf);
            OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height,␣
↪OEScale.AutoScale);
            oedepict.OESetup2DMolDisplayOptions(opts, itf);

            OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
            oedepict.OERenderMolecule(ofs, ext, disp);

            ofs.close();
        }
        catch (java.io.IOException e) {
            System.err.println("Unable to open interface file");
        }
    }
}
```

with interface file

```
# MDLQuery2Img interface file
!BRIEF [-in] <input> [-out] <output image>

!CATEGORY "input/output options :"

    !PARAMETER -in
      !ALIAS -i
      !TYPE string
      !REQUIRED true
      !KEYLESS 1
      !VISIBILITY simple
      !BRIEF Input filename
    !END

    !PARAMETER -out
      !ALIAS -o
      !TYPE string
      !REQUIRED true
      !KEYLESS 2
      !VISIBILITY simple
      !BRIEF Output filename
    !END
```

(continues on next page)

(continued from previous page)

```
!END
```

See also

- *MDL Query Depiction* chapter
- *OEConfigureImageOptions* function
- *OEConfigure2DMolDisplayOptions* function
- *OEIsRegisteredImageFile* function
- OEReadMDLQueryFile function
- *OEPrepareDepiction* function
- *OEGetImageWidth* and *OEGetImageHeight* functions
- *OE2DMolDisplayOptions* class
- *OESetup2DMolDisplayOptions* function
- *OE2DMolDisplay* class
- *OERenderMolecule* function

Examples

```
prompt> java MDLQuery2Img -in mdlquery.mol -out image.png
```

will generate the image shown in *Figure: Example of depicting an MDL query*

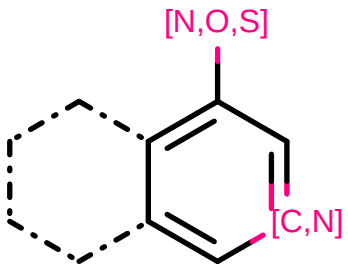


Fig. 4: **Example of depicting an MDL query**

See also

- *MDL Query Depiction* chapter

3.1.5 Depicting MDL Reaction

A program that converts an MDL reaction into an image file.

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> java MDLReaction2Img --help
```

will generate the following output:

```
Simple parameter list

 image options :
   -height : Height of output image
   -width : Width of output image

 input/output options :
   -in : Input filename
   -out : Output filename

 molecule display options :
   -aromstyle : Aromatic ring display style
   -atomcolor : Atom coloring style
   -atomlabelfontscale : Atom label font scale
   -atomprop : Atom property display
   -atomstereostyle : Atom stereo display style
   -bondcolor : Bond coloring style
   -bondprop : Bond property display
   -bondstereostyle : Bond stereo display style
   -hydrstyle : Hydrogen display style
   -linewidth : Default bond line width
   -protgroupdisp : Protective group display style
   -scale : Scaling of the depicted molecule
   -superdisp : Super atom display style
   -titleloc : Location of the molecule title
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
```

(continues on next page)

(continued from previous page)

```
/*****************************************************************************
 * Converts an MDL reaction into an image file.
 * The output file format depends on its file extension.
 *****************************************************************************/
package openeye.examples.oedepict;

import java.net.URL;
import java.util.*;

import openeye.oechem.*;
import openeye.oedepict.*;

public class MDLReaction2Img {

    public static void main(String argv[]) {
        URL fileURL = MDLReaction2Img.class.getResource("MDLReaction2Img.txt");
        try {
            OEInterface itf = new OEInterface();
            oechem.OEConfigureFromURL(itf, fileURL);
            oedepict.OEConfigureImageOptions(itf);
            oedepict.OEConfigure2DMolDisplayOptions(itf);

            if (!oechem.OEParseCommandLine(itf, argv, "MDLReaction2Img")) {
                oechem.OEThrow.Fatal("Unable to interpret command line!");
            }

            String iname = itf.GetString("-in");
            String oname = itf.GetString("-out");

            String ext = oechem.OEGetFileExtension(oname);
            if (!oedepict.OEIsRegisteredImageFile(ext)) {
                oechem.OEThrow.Fatal("Unknown image type!");
            }

            oemolistream ifs = new oemolistream();
            if (!ifs.open(iname)) {
                oechem.OEThrow.Fatal("Cannot open input file!");
            }

            oeofstream ofs = new oeofstream();
            if (!ofs.open(oname)) {
                oechem.OEThrow.Fatal("Cannot open output file!");
            }

            OEGraphMol mol = new OEGraphMol();
            if (!oechem.OEReadMDLReactionQueryFile(ifs, mol)) {
                oechem.OEThrow.Fatal("Cannot read mdl reaction!");
            }
            ifs.close();

            boolean suppressH   = false;
            boolean clearcoords = false;
```

(continues on next page)

(continued from previous page)

```
            oedepict.OEPrepareDepiction(mol, clearcoords, suppressH);

            double width  = oedepict.OEGetImageWidth (itf);
            double height = oedepict.OEGetImageHeight(itf);
            OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height,␣
↪OEScale.AutoScale);
            oedepict.OESetup2DMolDisplayOptions(opts, itf);

            OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
            oedepict.OERenderMolecule(ofs, ext, disp);

            ofs.close();
        }
        catch (java.io.IOException e) {
            System.err.println("Unable to open interface file");
        }
    }
}
```

with interface file

```
# MDLReaction2Img interface file
!BRIEF [-in] <input> [-out] <output image>

!CATEGORY "input/output options :"

    !PARAMETER -in
      !ALIAS -i
      !TYPE string
      !REQUIRED true
      !KEYLESS 1
      !VISIBILITY simple
      !BRIEF Input filename
    !END

    !PARAMETER -out
      !ALIAS -o
      !TYPE string
      !REQUIRED true
      !KEYLESS 2
      !VISIBILITY simple
      !BRIEF Output filename
    !END

!END
```

See also

- *MDL Reaction Depiction* chapter
- *OEConfigureImageOptions* function
- *OEConfigure2DMolDisplayOptions* function

- *OEIsRegisteredImageFile* function
- OEReadMDLReactionQueryFile function
- *OEPrepareDepiction* function
- *OEGetImageWidth* and *OEGetImageHeight* functions
- *OE2DMolDisplayOptions* class
- *OESetup2DMolDisplayOptions* function
- *OE2DMolDisplay* class
- *OERenderMolecule* function

Examples

```
prompt> java MDLReaction2Img -in mdlreaction.sdf -out image.png
```

will generate the following image: *Figure: Example of depicting an MDL reaction*

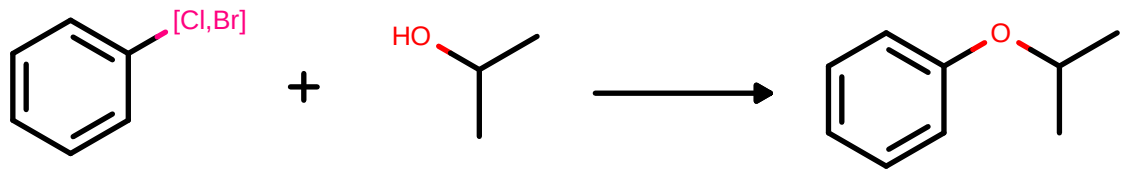


Fig. 5: **Example of depicting an MDL reaction**

```
prompt> java MDLReaction2img -in mdlreaction.sdf -out image.png -atomprop MapIdx
```

will generate the following image: *Figure: Example of depicting an MDL reaction with atom map indices*

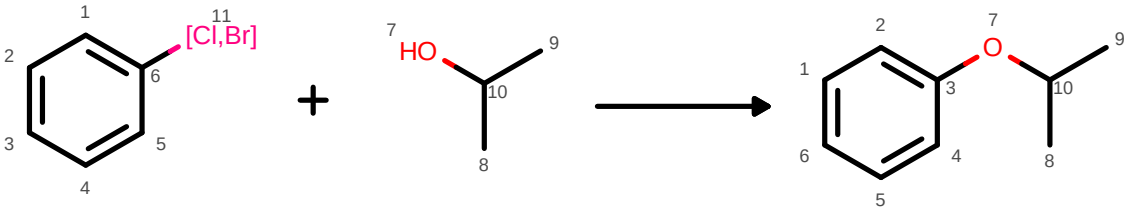


Fig. 6: **Example of depicting an MDL reaction with atom map indices**

See also

- *MDL Reaction Depiction* chapter

3.1.6 Depicting a Single Molecule

A program that converts a molecular structure into an image file.

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> java Mol2Img --help
```

will generate the following output:

```
Simple parameter list
 image options :
   -height : Height of output image
   -width : Width of output image

 input/output options :
   -in : Input filename
   -out : Output filename
   -ringdict : User-defined 2D ring dictionary

 molecule display options :
   -aromstyle : Aromatic ring display style
   -atomcolor : Atom coloring style
   -atomlabelfontscale : Atom label font scale
   -atomprop : Atom property display
   -atomstereostyle : Atom stereo display style
   -bondcolor : Bond coloring style
   -bondprop : Bond property display
   -bondstereostyle : Bond stereo display style
   -hydrstyle : Hydrogen display style
   -linewidth : Default bond line width
   -protgroupdisp : Protective group display style
   -scale : Scaling of the depicted molecule
   -superdisp : Super atom display style
   -titleloc : Location of the molecule title

 prepare depiction options :
   -clearcoords : Clear and regenerate 2D coordinates of molecule(s)
   -orientation : Set the preferred orientation of 2D coordinates
   -suppressH : Suppress explicit hydrogens of molecule(s)
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
```

(continues on next page)

(continued from previous page)

```
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
/*****************************************************************************
 * Converts a molecule structure into an image file.
 * The output file format depends on its file extension.
 *****************************************************************************/
package openeye.examples.oedepict;

import java.net.URL;
import java.util.*;

import openeye.oechem.*;
import openeye.oedepict.*;

public class Mol2Img {

    public static void main(String argv[]) {
        URL fileURL = Mol2Img.class.getResource("Mol2Img.txt");
        try {
            OEInterface itf = new OEInterface();
            oechem.OEConfigureFromURL(itf, fileURL);
            oedepict.OEConfigureImageOptions(itf);
            oedepict.OEConfigurePrepareDepictionOptions(itf);
            oedepict.OEConfigure2DMolDisplayOptions(itf);

            if (!oechem.OEParseCommandLine(itf, argv, "Mol2Img"))
                oechem.OEThrow.Fatal("Unable to interpret command line!");

            String iname = itf.GetString("-in");
            String oname = itf.GetString("-out");

            String ext = oechem.OEGetFileExtension(oname);
            if (!oedepict.OEIsRegisteredImageFile(ext))
                oechem.OEThrow.Fatal("Unknown image type!");

            oemolistream ifs = new oemolistream();
            if (!ifs.open(iname))
                oechem.OEThrow.Fatal("Cannot open input file!");
```

(continues on next page)

(continued from previous page)

```
            oeofstream ofs = new oeofstream();
            if (!ofs.open(oname))
                oechem.OEThrow.Fatal("Cannot open output file!");

            OEGraphMol mol = new OEGraphMol();
            if (!oechem.OEReadMolecule(ifs, mol))
                oechem.OEThrow.Fatal("Cannot read input file!");
            ifs.close();

            if (itf.HasString("-ringdict")) {
                String rdfname = itf.GetString("-ringdict");
                if (!oechem.OEInit2DRingDictionary(rdfname))
                    oechem.OEThrow.Warning("Cannot use user-defined ring dictionary!");
            }

            OEPrepareDepictionOptions popts = new OEPrepareDepictionOptions();
            oedepict.OESetupPrepareDepictionOptions(popts, itf);

            oedepict.OEPrepareDepiction(mol, popts);

            double width  = oedepict.OEGetImageWidth (itf);
            double height = oedepict.OEGetImageHeight(itf);
            OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height,␣
↪OEScale.AutoScale);
            oedepict.OESetup2DMolDisplayOptions(opts, itf);

            OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
            oedepict.OERenderMolecule(ofs, ext, disp);

            ofs.close();
        }
        catch (java.io.IOException e) {
            System.err.println("Unable to open interface file");
        }
    }
}
```

with interface file

```
# Mol2Img interface file
!BRIEF [-in] <input> [-out] <output image> [-ringdict] <rd file>

!CATEGORY "input/output options :"

  !PARAMETER -in
    !ALIAS -i
    !TYPE string
    !REQUIRED true
    !KEYLESS 1
    !VISIBILITY simple
    !BRIEF Input filename
  !END
```

(continues on next page)

(continued from previous page)

```
  !PARAMETER -out
    !ALIAS -o
    !TYPE string
    !REQUIRED true
    !KEYLESS 2
    !VISIBILITY simple
    !BRIEF Output filename
  !END

  !PARAMETER -ringdict
    !ALIAS -rd
    !TYPE string
    !REQUIRED false
    !VISIBILITY simple
    !BRIEF User-defined 2D ring dictionary
    !DETAIL
        2D ring dictionaries can be generated by the following OEChem examples:
        C++    - createringdict.cpp
        Python - createringdict.py
        Java   - CreateRingDict.java
        C#     - CreateRingDict.cs
  !END

!END
```

See also

- *Molecule Depiction* chapter
- *OEConfigureImageOptions* function
- *OEConfigurePrepareDepictionOptions* function
- *OEConfigure2DMolDisplayOptions* function
- *OEIsRegisteredImageFile* function
- *OEPrepareDepictionOptions* class
- *OESetupPrepareDepictionOptions* function
- *OEPrepareDepiction* function
- *OEGetImageWidth* and *OEGetImageHeight* functions
- *OE2DMolDisplayOptions* class
- *OESetup2DMolDisplayOptions* function
- *OE2DMolDisplay* class
- *OERenderMolecule* function

Examples

```
prompt> java Mol2Img -in nexium.ism -out image.png
```

will generate the image shown in *Figure: Example of using the program with default parameters*.

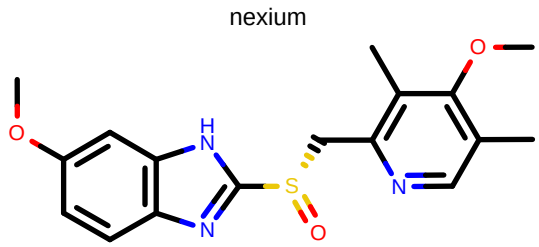


Fig. 7: **Example of using the program with default parameters**

```
prompt> java Mol2Img -in nexium.ism -out image.png -atomcolor WhiteMonochrome
```

will generate the image shown in *Figure: Example of using the program with the white monochrome atom coloring style*.

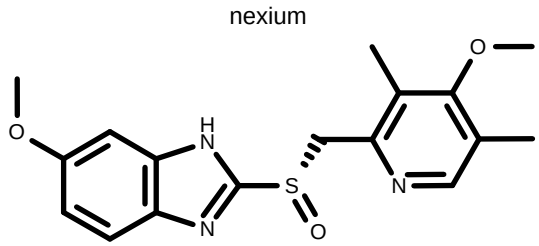


Fig. 8: **Example of using the program with the white monochrome atom coloring style**

```
prompt> java Mol2Img -in nexium.ism -out image.png -aromstyle Circle
```

will generate the image shown in *Figure: Example of using the program with aromaticity style 'Circle'*.

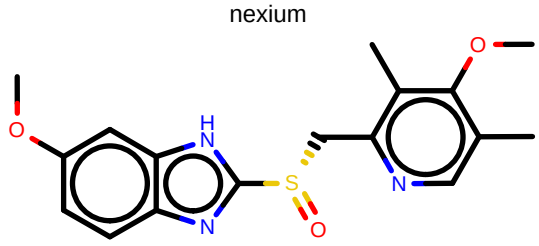


Fig. 9: **Example of using the program with aromaticity style 'Circle'**

```
prompt> java Mol2Img -in nexium.ism -out image.png -atomstereostyle
↪"AtomStereo|CIPAtomStereo"
```

will generate the image shown in *Figure: Example of the mol2img program with CIP atom stereo option*.

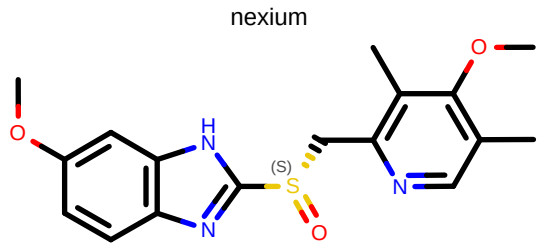


Fig. 10: **Example of using the program with CIP atom stereo option**

```
prompt> java Mol2Img -in nexium.ism -out image.png -linewidth 4
```

will generate the image shown in *Figure: Example of using the program with user-defined line width*.

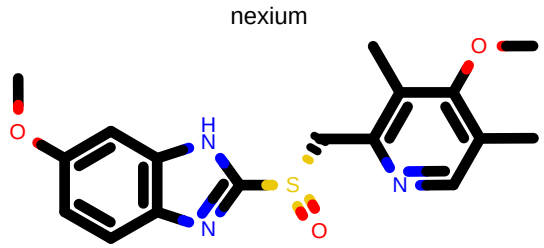


Fig. 11: **Example of using the program with user-defined line width**

```
prompt> java Mol2Img -in nexium.ism -out image.png -superdisp All
```

will generate the image shown in *Figure: Example of using the program with superatom style*.

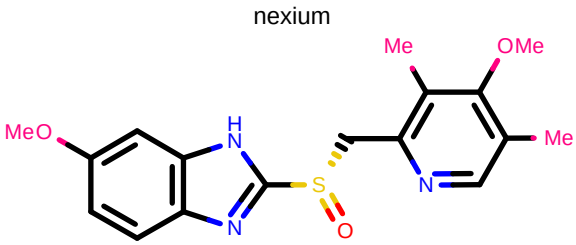


Fig. 12: **Example of using the program with superatom style**

Note

The above program can generate 2D coordinates with user-defined 2D ring layouts when using the **-ringdict** parameter.

See also the 2D Coordinate Generation Examples section of the **OEChem TK** manual that shows examples about how to generate and utilize user-defined ring dictionaries.

See also

- *Molecule Depiction* chapter

3.1.7 Depicting Molecules In a Grid

A program that converts molecular structures into an image with a grid layout. By default the input molecules are depicted in a 3x2 grid.

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> java Mols2Img --help
```

will generate the following output:

```
Simple parameter list
 -height : Height of output image
 -width : Width of output image

 image grid options :
   -cols : Number of columns
   -rows : Number of rows

 input/output options :
   -in : Input filename(s)
   -out : Output filename

 molecule display options :
   -aromstyle : Aromatic ring display style
   -atomcolor : Atom coloring style
   -atomlabelfontscale : Atom label font scale
   -atomprop : Atom property display
   -atomstereostyle : Atom stereo display style
   -bondcolor : Bond coloring style
   -bondprop : Bond property display
   -bondstereostyle : Bond stereo display style
   -hydrstyle : Hydrogen display style
   -linewidth : Default bond line width
   -protgroupdisp : Protective group display style
   -scale : Scaling of the depicted molecule
   -superdisp : Super atom display style
   -titleloc : Location of the molecule title

 prepare depiction options :
   -clearcoords : Clear and regenerate 2D coordinates of molecule(s)
   -orientation : Set the preferred orientation of 2D coordinates
   -suppressH : Suppress explicit hydrogens of molecule(s)
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
/*****************************************************************************
 * Converts molecules into an image with grid layout.
 * The output file format depends on its file extension.
 *****************************************************************************/
package openeye.examples.oedepict;

import java.net.URL;
import java.util.*;

import openeye.oechem.*;
import openeye.oedepict.*;

public class Mols2Img {

    public static void main(String argv[]) {
        URL fileURL = Mols2Img.class.getResource("Mols2Img.txt");
        try {
            OEInterface itf = new OEInterface();
            oechem.OEConfigureFromURL(itf, fileURL);
            oedepict.OEConfigureImageWidth (itf, 400.0);
            oedepict.OEConfigureImageHeight(itf, 400.0);
            oedepict.OEConfigureImageGridParams(itf);
            oedepict.OEConfigurePrepareDepictionOptions(itf);
            oedepict.OEConfigure2DMolDisplayOptions(itf);

            if (!oechem.OEParseCommandLine(itf, argv, "Mols2Img")) {
                oechem.OEThrow.Fatal("Unable to interpret command line!");
            }

            String oname = itf.GetString("-out");
            String ext = oechem.OEGetFileExtension(oname);
            if (!oedepict.OEIsRegisteredImageFile(ext)) {
                oechem.OEThrow.Fatal("Unknown image type!");
```

(continues on next page)

(continued from previous page)

```
            }

            oeofstream ofs = new oeofstream();
            if (!ofs.open(oname)) {
                oechem.OEThrow.Fatal("Cannot open output file!");
            }

            double width  = oedepict.OEGetImageWidth (itf);
            double height = oedepict.OEGetImageHeight(itf);
            OEImage image = new OEImage(width, height);

            int rows = oedepict.OEGetImageGridNumRows(itf);
            int cols = oedepict.OEGetImageGridNumColumns(itf);
            OEImageGrid grid = new OEImageGrid(image, rows, cols);

            OEPrepareDepictionOptions popts = new OEPrepareDepictionOptions();
            oedepict.OESetupPrepareDepictionOptions(popts, itf);

            OE2DMolDisplayOptions dopts = new OE2DMolDisplayOptions();
            oedepict.OESetup2DMolDisplayOptions(dopts, itf);
            dopts.SetDimensions(grid.GetCellWidth(), grid.GetCellHeight(), OEScale.
↪AutoScale);

            OEImageBaseIter celliter = grid.GetCells();
            for (String iname : itf.GetStringList("-in")) {
                oemolistream ifs = new oemolistream();
                if (!ifs.open(iname)) {
                    oechem.OEThrow.Warning("Cannot open " + iname + " input file!");
                    continue;
                }

                OEGraphMol mol = new OEGraphMol();
                while (oechem.OEReadMolecule(ifs, mol) && celliter.IsValid()) {
                    oedepict.OEPrepareDepiction(mol, popts);
                    OE2DMolDisplay disp = new OE2DMolDisplay(mol, dopts);
                    oedepict.OERenderMolecule(celliter.Target(), disp);
                    celliter.Increment();
                }
                ifs.close();
            }
            oedepict.OEWriteImage(ofs, ext, image);
            ofs.close();
        }
        catch (java.io.IOException e) {
            System.err.println("Unable to open interface file");
        }
    }
}
```

with interface file

```
# Mols2Img interface file
```

(continues on next page)

(continued from previous page)

```
!BRIEF -in <input1> [ <input2> .. ] -out <output image>

!CATEGORY "input/output options :"

    !PARAMETER -in
      !ALIAS -i
      !TYPE string
      !LIST true
      !REQUIRED true
      !VISIBILITY simple
      !BRIEF Input filename(s)
    !END

    !PARAMETER -out
      !ALIAS -o
      !TYPE string
      !REQUIRED true
      !VISIBILITY simple
      !BRIEF Output filename
    !END

!END
```

See also

- *Molecule Layouts* chapter
- *OEConfigureImageWidth* function
- *OEConfigureImageHeight* function
- *OEConfigureImageGridParams* function
- *OEConfigurePrepareDepictionOptions* function
- *OEConfigure2DMolDisplayOptions* function
- *OEGetImageWidth* and *OEGetImageHeight* functions
- *OEGetImageGridNumRows* and *OEGetImageGridNumColumns* functions
- *OEImageGrid* class
- *OEPrepareDepictionOptions* class
- *OESetupPrepareDepictionOptions* function
- *OE2DMolDisplayOptions* class
- *OESetup2DMolDisplayOptions* function
- *OEPrepareDepiction* function
- *OERenderMolecule* function
- *OEWriteImage* function

Examples

```
prompt> java Mols2Img -in amino.ism -out image.png
```

will generate the image shown in *Figure: Example of using the program with default parameters*.

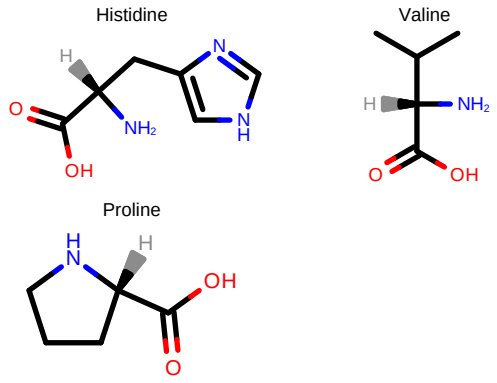


Fig. 13: **Example of using the program with default parameters**

Note

By default the program depicts molecules in a 3x2 grid

```
prompt> java Mols2Img -in amino.ism -out image.png -rows 2 -cols 3
```

will generate the image shown in *Figure: Example of using the program with user-defined number of rows and columns*.

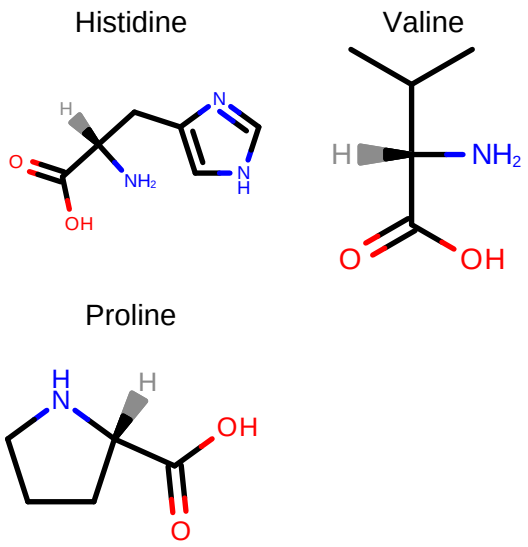


Fig. 14: **Example of using the program with user-defined number of rows and columns**

```
prompt> java Mols2Img -in amino.ism -out image.png  -rows 2 -cols 2 -titleloc Bottom
```

will generate the image shown in *Figure: Example of using the program with molecule titles displayed at the bottom*.

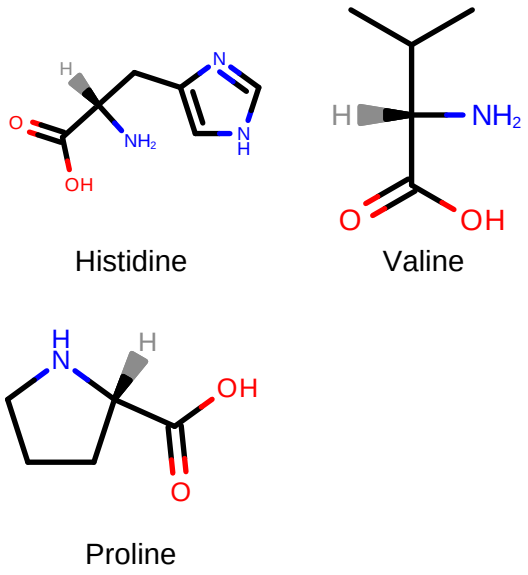


Fig. 15: **Example of using the program with molecule titles displayed at the bottom**

See also

- *Depicting Molecules in a Grid* section

3.1.8 Depicting Molecules in Multi-Page

A program that converts molecular structures into a multipage PDF image.

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> java Mols2PDF --help
```

will generate the following output:

```
Simple parameter list
input/output options :
  -in : Input filename
  -out : Output filename
  -ringdict : User-defined 2D ring dictionary

molecule display options :
  -aromstyle : Aromatic ring display style
  -atomcolor : Atom coloring style
  -atomlabelfontscale : Atom label font scale
  -atomprop : Atom property display
  -atomstereostyle : Atom stereo display style
  -bondcolor : Bond coloring style
  -bondprop : Bond property display
  -bondstereostyle : Bond stereo display style
  -hydrstyle : Hydrogen display style
  -linewidth : Default bond line width
  -protgroupdisp : Protective group display style
  -scale : Scaling of the depicted molecule
  -superdisp : Super atom display style
  -titleloc : Location of the molecule title

prepare depiction options :
  -clearcoords : Clear and regenerate 2D coordinates of molecule(s)
  -orientation : Set the preferred orientation of 2D coordinates
  -suppressH : Suppress explicit hydrogens of molecule(s)

report options :
  -colsperpage : Number of columns per page
  -pageheight : Page height
  -pageorientation : Page orientation
  -pagesize : Page size
  -pagewidth : Page width
  -rowsperpage : Number of rows per page
```

See also

- *Multi Page Reports* chapter

Note

The above program can generate 2D coordinates with user-defined 2D ring layouts when using the **-ringdict** parameter.

See also the 2D Coordinate Generation Examples section of the **OEChem TK** manual that shows examples about how to generate and utilize user-defined ring dictionaries.

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
/*****************************************************************************
 * Converts molecules into a multi-page PDF document.
 *****************************************************************************/
package openeye.examples.oedepict;

import java.net.URL;
import java.util.*;

import openeye.oechem.*;
import openeye.oedepict.*;

public class Mols2PDF {

    public static void main(String argv[]) {
        URL fileURL = Mols2PDF.class.getResource("Mols2PDF.txt");
        try {
            OEInterface itf = new OEInterface();
            oechem.OEConfigureFromURL(itf, fileURL);
            oedepict.OEConfigureReportOptions(itf);
            oedepict.OEConfigurePrepareDepictionOptions(itf);
            oedepict.OEConfigure2DMolDisplayOptions(itf);

            if (!oechem.OEParseCommandLine(itf, argv, "Mols2PDF")) {
                return;
            }
```

(continues on next page)

(continued from previous page)

```
            String iname = itf.GetString("-in");
            oemolistream ifs = new  oemolistream();
            if (!ifs.open(iname)) {
                oechem.OEThrow.Fatal("Cannot open" + iname + " input file!");
            }

            String oname = itf.GetString("-out");
            String ext = oechem.OEGetFileExtension(oname);
            if (!ext.equals("pdf")) {
                oechem.OEThrow.Fatal("Output must be PDF format.");
            }

            oeofstream ofs = new oeofstream();
            if (!ofs.open(oname)) {
                oechem.OEThrow.Fatal("Cannot open " + oname + " output file!");
            }

            if (itf.HasString("-ringdict")) {
                String rdfname = itf.GetString("-ringdict");
                if (!oechem.OEInit2DRingDictionary(rdfname))
                    oechem.OEThrow.Warning("Cannot use user-defined ring dictionary!");
            }

            OEReportOptions ropts = new OEReportOptions();
            oedepict.OESetupReportOptions(ropts, itf);
            ropts.SetFooterHeight(25.0);
            OEReport report = new OEReport(ropts);

            OEPrepareDepictionOptions popts = new OEPrepareDepictionOptions();
            oedepict.OESetupPrepareDepictionOptions(popts, itf);

            OE2DMolDisplayOptions dopts = new OE2DMolDisplayOptions();
            oedepict.OESetup2DMolDisplayOptions(dopts, itf);
            dopts.SetDimensions(report.GetCellWidth(), report.GetCellHeight(), OEScale.
↪AutoScale);

            OEGraphMol mol = new OEGraphMol();
            while (oechem.OEReadMolecule(ifs, mol)) {
                OEImageBase cell = report.NewCell();
                oedepict.OEPrepareDepiction(mol, popts);
                OE2DMolDisplay disp = new OE2DMolDisplay(mol, dopts);
                oedepict.OERenderMolecule(cell, disp);
            }
            ifs.close();

            int pagenum = 1;
            OEFont font = new OEFont(OEFontFamily.Default, OEFontStyle.Bold, 12,␣
↪OEAlignment.Center, oechem.getOEBlack());
            for (OEImageBase footer : report.GetFooters()) {
                String text = "Page " + Integer.toString(pagenum) + " of " + Integer.
↪toString(report.NumPages());
```

(continues on next page)

(continued from previous page)

```
                oedepict.OEDrawTextToCenter(footer, text, font);
                pagenum += 1;
            }
            oedepict.OEWriteReport(ofs, ext, report);
            ofs.close();

        }
        catch (java.io.IOException e) {
            System.err.println("Unable to open interface file");
        }
    }
}
```

with interface file

```
# Mols2PDF interface file
!BRIEF [-in] <input> [-out] <output pdf> [-ringdict] <rd file>

!CATEGORY "input/output options :"

  !PARAMETER -in
    !ALIAS -i
    !TYPE string
    !REQUIRED true
    !KEYLESS 1
    !VISIBILITY simple
    !BRIEF Input filename
  !END

  !PARAMETER -out
    !ALIAS -o
    !TYPE string
    !REQUIRED true
    !KEYLESS 2
    !VISIBILITY simple
    !BRIEF Output filename
  !END

  !PARAMETER -ringdict
    !ALIAS -rd
    !TYPE string
    !REQUIRED false
    !VISIBILITY simple
    !BRIEF User-defined 2D ring dictionary
    !DETAIL
        2D ring dictionaries can be generated by the following OEChem examples:
        C++    - createringdict.cpp
        Python - createringdict.py
        Java   - CreateRingDict.java
        C#     - CreateRingDict.cs
  !END
```

(continues on next page)

(continued from previous page)

```
!END
```

See also

- *Multi Page Images* chapter
- *OEConfigureReportOptions* function
- *OEConfigurePrepareDepictionOptions* function
- *OEConfigure2DMolDisplayOptions* function
- *OEReportOptions* class
- *OESetupReportOptions* function
- *OEReport* class
- *OEPrepareDepictionOptions* class
- *OESetupPrepareDepictionOptions* function
- *OE2DMolDisplayOptions* class
- *OESetup2DMolDisplayOptions* function
- *OEPrepareDepiction* function
- *OE2DMolDisplay* class
- *OERenderMolecule* function
- *OEFont* class
- *OEDrawTextToCenter* function
- *OEWriteReport* function

3.1.9 Generating 2D Ring Dictionary Report

A program that generate a multi-page PDF report of the 2D ring dictionary.

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> java RingDict2PDF --help
```

Will generate the following output:

```
Simple parameter list
    input/output options :
      -out : Output filename
      -ringdict : Input ring dictionary filename

    report options :
      -colsperpage : Number of columns per page
      -pageheight : Page height
      -pageorientation : Page orientation
```

(continues on next page)

(continued from previous page)

```
    -pagesize : Page size
    -pagewidth : Page width
    -rowsperpage : Number of rows per page
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
/*****************************************************************************
 * Converts a ring dictionary into a multi-page PDF document.
 *****************************************************************************/
package openeye.examples.oedepict;

import java.net.URL;
import java.util.*;

import openeye.oechem.*;
import openeye.oedepict.*;

public class RingDict2PDF {

    public static void main(String argv[]) {
        URL fileURL = RingDict2PDF.class.getResource("RingDict2PDF.txt");
        try {
            OEInterface itf = new OEInterface();
            oechem.OEConfigureFromURL(itf, fileURL);
            oedepict.OEConfigureReportOptions(itf);

            if (!oechem.OEParseCommandLine(itf, argv, "RingDict2PDF"))
                oechem.OEThrow.Fatal("Unable to interpret command line!");

            String ifname = itf.GetString("-ringdict");
            String ofname = itf.GetString("-out");

            if (!oechem.OEIsValid2DRingDictionary(ifname))
```

(continues on next page)

(continued from previous page)

```
                oechem.OEThrow.Fatal("Invalid ring dictionary file!");

            OE2DRingDictionary ringdict = new OE2DRingDictionary(ifname);

            OEReportOptions ropts = new OEReportOptions();
            oedepict.OESetupReportOptions(ropts, itf);

            oedepict.OEWrite2DRingDictionaryReport(ofname, ringdict, ropts);

        }
        catch (java.io.IOException e) {
            System.err.println("Unable to open interface file");
        }
    }
}
```

with interface file

```
# RingDict2PDF interface file
!BRIEF [-ringdict] <input ringdict> [-out] <output pdf>

!CATEGORY "input/output options :"

    !PARAMETER -ringdict
      !ALIAS -rd
      !TYPE string
      !REQUIRED true
      !KEYLESS 1
      !VISIBILITY simple
      !BRIEF Input ring dictionary filename
      !DETAIL
        2D ring dictionaries can be generated by the following OEChem examples:
        C++    - createringdict.cpp
        Python - createringdict.py
        Java   - CreateRingDict.java
        C#     - CreateRingDict.cs
    !END

    !PARAMETER -out
      !ALIAS -o
      !TYPE string
      !REQUIRED true
      !KEYLESS 2
      !VISIBILITY simple
      !BRIEF Output filename
    !END

!END
```

See also

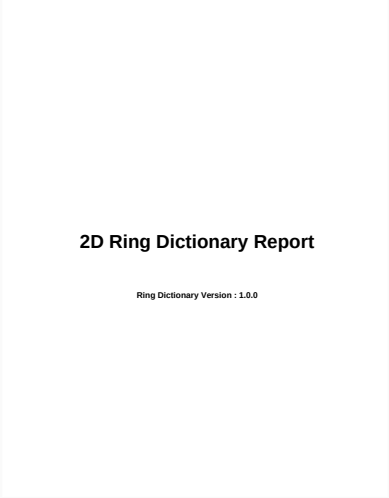
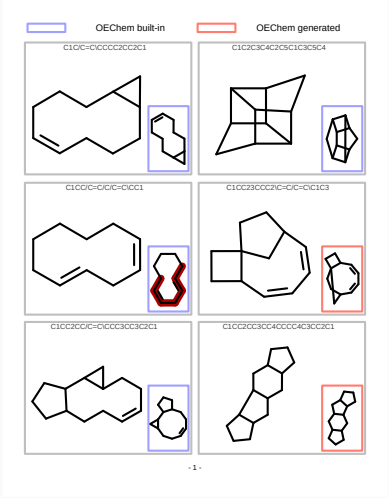
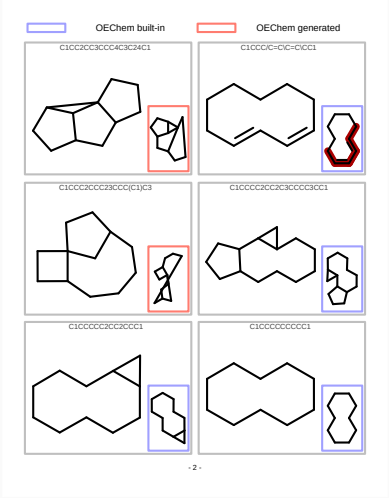
- 2D Coordinate Generation Examples section in the **OEChem TK** manual
- OE2DRingDictionary class
- `OEIsValid2DRingDictionary` function
- *`OEWrite2DRingDictionaryReport`* function

Example

```
prompt> java RingDict2PDF -ringdict rtd.oeb -out rtd.pdf
```

will generate a multi-page PDF report similar to the one shown in *Table: Example of ring dictionary report*.

Table 2: **Example of ring dictionary report** (The pages are reduced here for visualization convenience)

cover page	page 1	page 2
		

See also

- OE2DRingDictionary class in the **OEChem TK** manual
- 2D Coordinate Generation Examples section in the **OEChem TK** manual that shows examples about how to create and manipulate user-defined ring dictionaries.

3.1.10 Aligning Molecules Based on Substructure Search

A program that aligns the molecules based on a substructure search initialized with the given SMARTS pattern.

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> java SMARTSAlign2D --help
```

Will generate the following output:

```
Simple parameter list
input/output options :
   -in : Filename of input molecules
   -out : Output filename
   -smarts : SMARTS for alignment match
```

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
/*****************************************************************************
 *  Aligns molecules to a smarts pattern.
 *****************************************************************************/
package openeye.examples.oedepict;

import java.io.IOException;
import java.net.URL;
import java.util.*;

import openeye.oechem.*;
import openeye.oedepict.*;

public class SMARTSAlign2D {

    public static void main(String argv[]) {
        URL fileURL = SMARTSAlign2D.class.getResource("SMARTSAlign2D.txt");
        OEInterface itf = null;
        try {
            itf = new OEInterface(fileURL, "SMARTSAlign2D", argv);
        }
        catch (IOException e) {
            oechem.OEThrow.Fatal("Unable to open interface file");
        }

        String iname = itf.GetString("-in");
```

(continues on next page)

(continued from previous page)

```
        String oname = itf.GetString("-out");
        String smarts = itf.GetString("-smarts");

        OEQMol qmol = new OEQMol();
        if (!oechem.OEParseSmarts(qmol, smarts))
            oechem.OEThrow.Fatal("Invalid SMARTS: " + smarts);

        oechem.OEGenerate2DCoordinates(qmol);

        OESubSearch ss = new OESubSearch(qmol);
        if (!ss.IsValid())
            oechem.OEThrow.Fatal("Unable to initialize substructure search!");

        oemolistream ifs = new oemolistream();
        if (!ifs.open(iname))
            oechem.OEThrow.Fatal("Cannot open reference molecule file!");

        oemolostream ofs = new oemolostream();
        if (!ofs.open(oname))
            oechem.OEThrow.Fatal("Cannot open output file!");
        if (!oechem.OEIs2DFormat(ofs.GetFormat()))
            oechem.OEThrow.Fatal("Invalid output format for 2D coordinates");

        OEGraphMol mol = new OEGraphMol();
        while (oechem.OEReadMolecule(ifs, mol)) {
            oechem.OEPrepareSearch(mol, ss);

            OEAlignmentResult alignres = new OEAlignmentResult(oedepict.
↪OEPrepareAlignedDepiction(mol, ss));
            if (!alignres.IsValid())
            {
                oechem.OEThrow.Warning("Substructure is not found in input molecule!");
                oedepict.OEPrepareDepiction(mol);
            }
            oechem.OEWriteMolecule(ofs, mol);
        }
        ifs.close();
        ofs.close();
    }
}
```

with interface file

```
# SMARTSAlign2D interface file
!BRIEF [-in] <input> [-smarts] <smarts> [-out] <output>

!CATEGORY "input/output options :"

    !PARAMETER -in
      !ALIAS -i
      !TYPE string
      !REQUIRED true
```

(continues on next page)

(continued from previous page)

```
      !KEYLESS 1
      !VISIBILITY simple
      !BRIEF Filename of input molecules
    !END

    !PARAMETER -smarts
      !ALIAS -s
      !TYPE string
      !REQUIRED true
      !KEYLESS 2
      !VISIBILITY simple
      !BRIEF SMARTS for alignment match
    !END

    !PARAMETER -out
      !ALIAS -o
      !TYPE string
      !REQUIRED true
      !KEYLESS 3
      !VISIBILITY simple
      !BRIEF Output filename
    !END

!END
```

See also

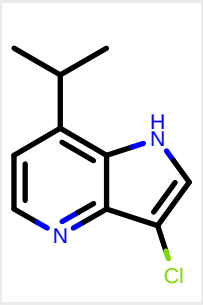
- *Molecule Alignment* chapter
- OESubSearch class
- `OEGenerate2DCoordinates` function
- `OEIs2DFormat` function
- `OEPrepareSearch` function
- *`OEPrepareAlignedDepiction`* function
- *`OEPrepareDepiction`* function

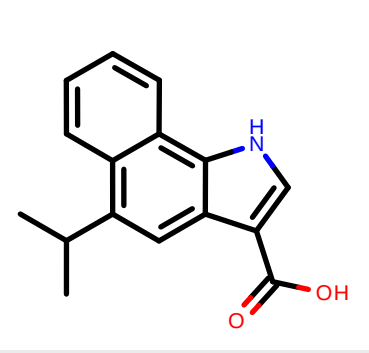
Examples

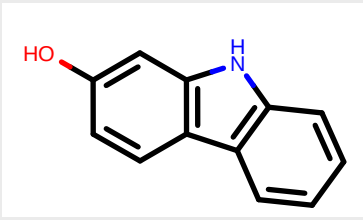
```
prompt> java SMARTSAlign2D -in test.sdf -out aligned.sdf -smarts "c1cc[c,n]c2c1ncc2" -
↪out out.mol
```

Taking the output coordinates and generating the images with the *mol2img* program will produce the images shown in *Table: Example of using the program to align molecule*

Table 3: **Example of using the program to align molecules**







See also

- *Molecule Alignment* chapter

3.1.11 Depicting MDL Query Substructure Search Hits

A program that performs substructure search on a molecule using an `MDL` query and generates an image file one match per molecule.

Command Line Interface

A description of the command line interface can be obtained by executing the program with the –help argument.

```
prompt> java ViewMDLSearch --help
```

Will generate the following output:

```
Simple parameter list
 general options :
   -align : Align hits to query

 highlight options :
   -highlightcolor : Highlighting color
   -highlightstyle : Highlighting style

 input/output options :
   -out : Output image filename
   -query : Input query filename
   -target : Input target filename

 molecule display options :
   -aromstyle : Aromatic ring display style
   -atomcolor : Atom coloring style
   -atomlabelfontscale : Atom label font scale
   -atomprop : Atom property display
   -atomstereostyle : Atom stereo display style
   -bondcolor : Bond coloring style
   -bondprop : Bond property display
```

(continues on next page)

(continued from previous page)

```
   -bondstereostyle : Bond stereo display style
   -hydrstyle : Hydrogen display style
   -linewidth : Default bond line width
   -protgroupdisp : Protective group display style
   -scale : Scaling of the depicted molecule
   -superdisp : Super atom display style
   -titleloc : Location of the molecule title

 report options :
   -colsperpage : Number of columns per page
   -pageheight : Page height
   -pageorientation : Page orientation
   -pagesize : Page size
   -pagewidth : Page width
   -rowsperpage : Number of rows per page
```

See also

- *Multi Page Reports* chapter

Code

```
/*
(C) 2022 Cadence Design Systems, Inc. (Cadence)
All rights reserved.
TERMS FOR USE OF SAMPLE CODE The software below ("Sample Code") is
provided to current licensees or subscribers of Cadence products or
SaaS offerings (each a "Customer").
Customer is hereby permitted to use, copy, and modify the Sample Code,
subject to these terms. Cadence claims no rights to Customer's
modifications. Modification of Sample Code is at Customer's sole and
exclusive risk. Sample Code may require Customer to have a then
current license or subscription to the applicable Cadence offering.
THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED.  OPENEYE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT
NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be
liable for any damages or liability in connection with the Sample Code
or its use.
*/
/*****************************************************************************
 * Performs substructure search on a molecule using a MDL query and
 * generates an image file depicting one match per molecule.
 * The output file format depends on its file extension.
 *****************************************************************************/
package openeye.examples.oedepict;

import java.net.URL;
import java.util.*;
```

(continues on next page)

(continued from previous page)

```
import openeye.oechem.*;
import openeye.oedepict.*;

public class ViewMDLSearch {

    public static void main(String argv[]) {
        URL fileURL = ViewMDLSearch.class.getResource("ViewMDLSearch.txt");
        try {
            OEInterface itf = new OEInterface();
            oechem.OEConfigureFromURL(itf, fileURL);
            oedepict.OEConfigureReportOptions(itf);
            oedepict.OEConfigure2DMolDisplayOptions(itf);
            oedepict.OEConfigureHighlightParams(itf);

            if (!oechem.OEParseCommandLine(itf, argv, "ViewMDLSearch"))
                oechem.OEThrow.Fatal("Unable to interpret command line!");

            String qname = itf.GetString("-query");
            String tname = itf.GetString("-target");
            String oname = itf.GetString("-out");

            String ext = oechem.OEGetFileExtension(oname);
            if (!oedepict.OEIsRegisteredMultiPageImageFile(ext)) {
                oechem.OEThrow.Fatal("Unknown multipage image type!");
            }

            oemolistream qfile = new oemolistream();
            if (!qfile.open(qname)) {
                oechem.OEThrow.Fatal("Cannot open mdl query file!");
            }
            if (qfile.GetFormat() != OEFormat.MDL && qfile.GetFormat() != OEFormat.SDF) {
                oechem.OEThrow.Fatal("Query file has to be an MDL file!");
            }

            oemolistream ifs = new  oemolistream();
            if (!ifs.open(tname)) {
                oechem.OEThrow.Fatal("Cannot open target input file!");
            }

            OEGraphMol depictquery = new OEGraphMol();
            if (!oechem.OEReadMDLQueryFile(qfile, depictquery)) {
                oechem.OEThrow.Fatal("Cannot read query molecule!");
            }
            qfile.close();

            oedepict.OEPrepareDepiction(depictquery);
            int queryopts = OEMDLQueryOpts.Default | OEMDLQueryOpts.SuppressExplicitH;
            OEQMol qmol = new OEQMol();
            oechem.OEBuildMDLQueryExpressions(qmol, depictquery, queryopts);

            OESubSearch ss = new OESubSearch();
            if (!ss.Init(qmol)) {
```

(continues on next page)

(continued from previous page)

```
            oechem.OEThrow.Fatal("Cannot initialize substructure search!");
        }

        int  hstyle = oedepict.OEGetHighlightStyle(itf);
        OEColor  hcolor = oedepict.OEGetHighlightColor(itf);
        boolean align  = itf.GetBool("-align");

        OEReportOptions ropts = new OEReportOptions();
        oedepict.OESetupReportOptions(ropts, itf);
        ropts.SetHeaderHeight(140.0);
        OEReport report = new OEReport(ropts);

        OE2DMolDisplayOptions dopts = new  OE2DMolDisplayOptions();
        oedepict.OESetup2DMolDisplayOptions(dopts, itf);
        dopts.SetDimensions(report.GetCellWidth(), report.GetCellHeight(), OEScale.
↪AutoScale);

        boolean unique = true;
        OEGraphMol mol = new OEGraphMol();
        while (oechem.OEReadMolecule(ifs, mol)) {
            oechem.OEPrepareSearch(mol, ss);
            Iterator<OEMatchBase> miter = ss.Match(mol, unique);
            if (miter != null && miter.hasNext()) {
                OEMatchBase match = miter.next();

                OEAlignmentResult alignres = new OEAlignmentResult(match);
                if (align)
                    alignres = oedepict.OEPrepareAlignedDepiction(mol, ss);
                else
                    oedepict.OEPrepareDepiction(mol);

                OEImageBase cell = report.NewCell();
                OE2DMolDisplay disp = new OE2DMolDisplay(mol, dopts);
                if (alignres.IsValid())
                    oedepict.OEAddHighlighting(disp, hcolor, hstyle, alignres);
                oedepict.OERenderMolecule(cell, disp);
                oedepict.OEDrawBorder(cell, oedepict.getOELightGreyPen());
            }
        }
        ifs.close();

        // render query structure in each header
        dopts.SetDimensions(report.GetHeaderWidth(), report.GetHeaderHeight(),␣
↪OEScale.AutoScale);
        OE2DMolDisplay disp = new OE2DMolDisplay(depictquery, dopts);
        for (OEImageBase header : report.GetHeaders()) {
            oedepict.OERenderMolecule(header, disp);
            oedepict.OEDrawBorder(header, oedepict.getOELightGreyPen());
        }

        oedepict.OEWriteReport(oname, report);
    }
```

(continues on next page)

(continued from previous page)

```
        catch (java.io.IOException e) {
            System.err.println("Unable to open interface file");
        }
    }
}
```

with interface file

```
# ViewMDLSearch interface file
!BRIEF [-query] <input> [-target] <input> [-out] <output multipage image> [-align]

!CATEGORY "input/output options :"

  !PARAMETER -query
    !ALIAS -q
    !TYPE string
    !REQUIRED true
    !KEYLESS 1
    !VISIBILITY simple
    !BRIEF Input query filename
  !END

  !PARAMETER -target
    !ALIAS -t
    !TYPE string
    !REQUIRED true
    !KEYLESS 2
    !VISIBILITY simple
    !BRIEF Input target filename
  !END

  !PARAMETER -out
    !ALIAS -o
    !TYPE string
    !REQUIRED true
    !KEYLESS 3
    !VISIBILITY simple
    !BRIEF Output image filename
  !END

!END

!CATEGORY "general options :"

  !PARAMETER -align
    !TYPE bool
    !REQUIRED false
    !DEFAULT false
    !VISIBILITY simple
    !BRIEF Align hits to query
  !END
```

(continues on next page)

(continued from previous page)

```
!END
```

See also

- *Multi Page Images* and *MDL Query Depiction* chapters
- *OEConfigureReportOptions* function
- *OEConfigure2DMolDisplayOptions* function
- *OEConfigureHighlightParams* function
- *OEIsRegisteredMultiPageImageFile* function
- OEReadMDLQueryFile function
- *OEPrepareDepiction* function
- OEBuildMDLQueryExpressions function
- OESubSearch class
- *OEGetHighlightStyle* and *OEGetHighlightColor* functions
- *OEReportOptions* class
- *OESetupReportOptions* function
- *OEReport* class
- *OE2DMolDisplayOptions* class
- *OESetup2DMolDisplayOptions* function
- OEPrepareSearch function
- *OEPrepareAlignedDepiction* function
- *OEPrepareDepiction* function
- *OE2DMolDisplay* class
- *OEAddHighlighting* function
- *OERenderMolecule* function
- *OEDrawBorder* function
- *OEWriteReport* function

The following Java example demonstrates the usage of *OEJavaImageBase* class.

Download code

`JavaImageBase.java`

3.2 Python Cookbook Examples

There are large number of code examples available in OpenEye Python Cookbook that use **OpenEye Toolkits** to solve a wide range of cheminformatics and molecular modeling problems.

- 2D Depiction chapter contains examples that illustrate how the **OEDepict TK** and **Grapheme TK** can be utilized to depict molecules and visualize properties calculated with other OpenEye toolkits.
- Visualizing 3D Information chapter contains examples that illustrate how the **Grapheme TK** can be utilized to project complex 3D information into the 2D molecular graph.
- Cheminformatics chapter contains examples that solve various cheminformatics problems such as similarity search, ring perception and manipulating molecular graphs.
- Peptide Informatics chapter contains examples that solve various peptide informatics problems such as generating custom monomer sets, parsing and generating HELM representations, and visualizing peptides.

OpenEye Python Cookbook examples using OEDepict TK

- Depicting Reaction Components
- Depicting Multiple Matches
- Depicting Library Generation
- Depicting Molecular Graph Symmetry
- Highlighting Fragments

CHAPTER

FOUR

API

4.1 OEDepict Classes

4.1.1 OE2DAtomDisplay

```
class OE2DAtomDisplay : public OEAtomDisplayBase
```

This class stores 2D depiction information of an atom. See *Figure: OEDepict TK atom display class hierarchy*.

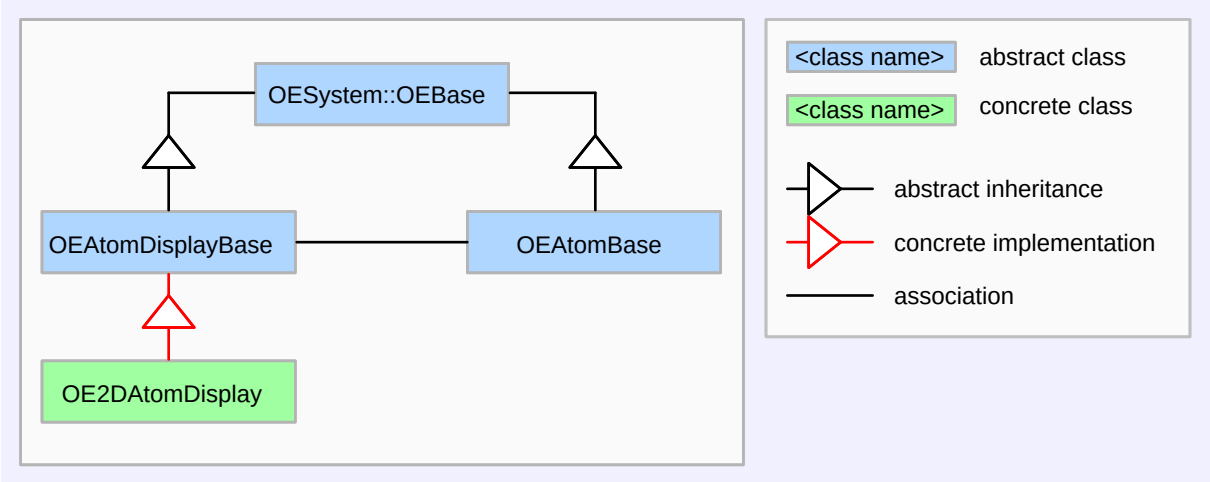


Fig. 1: **OEDepict TK atom display class hierarchy**

The following figure (*Figure: Example of the layout of an atom*) illustrates how atoms are rendered in the **OEDepict TK**.

- The atom label is positioned at the coordinates returned by the *OE2DAtomDisplay.GetCoords* method. The string that is rendered as an atom label is returned by the *OE2DAtomDisplay.GetLabel* method.
- If the atom has a charge, then it is rendered as a super-script on the right side of the atomic label.

> **See also**
>
> – *OE2DAtomDisplay.GetCharge* method

- If the atom has isotope information, then it is rendered as a super-script on the left side of the atomic label.

See also

- *OE2DAtomDisplay.GetMass* method

- If the atom has implicit hydrogen(s), then the *hydrogen label* is positioned in a cardinal direction relative to the *atomic label*. The number of implicit hydrogens and the position can be accessed by calling the *OE2DAtomDisplay.GetHCount* and the *OE2DAtomDisplay.GetHPosition* methods, respectively.

See also

- *OEHydrogenStyle* namespace
- *OEHydrogenPos* namespace

- All labels (isotope, atomic, charge and explicit hydrogen) has the same color that is associated with the atomic number.

See also

- *OEAtomColorStyle* namespace
- *Appendix: Element coloring (CPK)* chapter

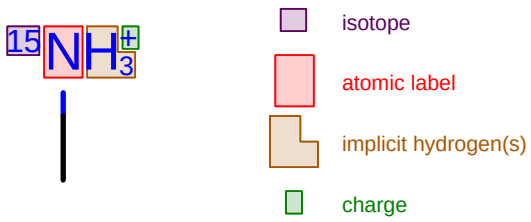


Fig. 2: **Example of the layout of an atom**

The following methods are publicly inherited from *OEAtomDisplayBase*:

GetDataType	*IsDataType*	*SetVisible*
GetAtom	*IsVisible*	

The *OE2DAtomDisplay* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
rendered formal charge	*GetCharge*	*SetCharge*	integer
position of the atom label	*GetCoords*	*SetCoords*	*OE2DPoint* class
hydrogen count	*GetHCount*	*SetHCount*	unsigned int
position of hydrogen label	*GetHPosition*	*SetHPosition*	*OEHydrogenPos* namespace
rendered atom label	*GetLabel*	*SetLabel*	string
font of atom label	*GetLabelFont*	*SetLabelFont*	*OEFont* class
rendered isotope number	*GetMass*	*SetMass*	integer
rendered string as property	*GetProperty*	*SetProperty*	string
font of property string	*GetPropertyFont*	*SetPropertyFont*	*OEFont* class
position of property label	*GetPropertyOffset*	*SetPropertyOffset*	*OE2DPoint* class
rendered radical type	*GetRadical*	*SetRadical*	*OERadicalDisplayType* namespace

Constructors

```
OE2DAtomDisplay(const OEChem::OEAtomBase* atom)
```

Construct an *OE2DAtomDisplay* object that contains the style information for `atom`.

```
OE2DAtomDisplay(const OE2DAtomDisplay& rhs)
```

Copy constructor.

CreateCopy

```
OESystem::OEBase* CreateCopy() const
```

Deep copy constructor that returns a pointer to a copy of the object. The memory returned is dynamically allocated and owned by the caller.

GetCharge

```
int GetCharge() const
```

Returns the charge rendered next to the atomic label. See example in *Figure: Example of charge representation*. The number will be rendered in the following ways:

0
: nothing is rendered

-1
: negative sign "-" is rendered

1
: positive sign "+" is rendered

less than -1
"-<n>" is rendered

greater than 1
"+<n>" is rendered

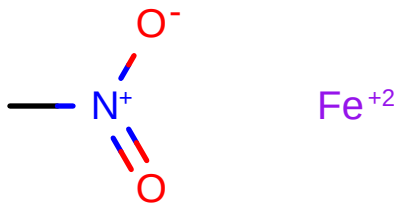


Fig. 3: **Example of charge representation**

See also

- *OE2DAtomDisplay.SetCharge* method

GetCoords

```
const OE2DPoint& GetCoords() const
```

Returns the coordinates of the atom in the coordinate system that is defined when a *OE2DMolDisplay* object is constructed with a specific width and height. This is the exact location of the atom when it is rendered into an image.

See also

- *OE2DAtomDisplay.SetCoords* method
- *OE2DPoint* method

GetHCount

```
unsigned int GetHCount() const
```

Returns the hydrogen count to render for this atom. See example in *Figure: Example of hydrogen count representation*. The number will be rendered in the following ways:

0
no "H" label is rendered

1
"H" is rendered next to the atomic label

greater than 1
this number is rendered as a sub-script of the "H" label

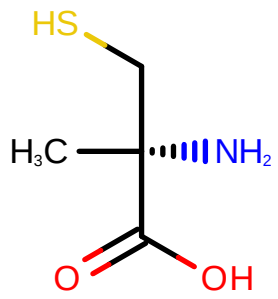


Fig. 4: **Example of hydrogen count representation**

See also

- *OE2DAtomDisplay.SetHCount* method
- *OE2DAtomDisplay.SetHPosition* method

GetHPosition

```
unsigned int GetHPosition() const
```

Returns the position of the hydrogen label relative to the atomic label. The return value is taken from the *OEHydrogenPos* namespace.

See also

- *OE2DAtomDisplay.SetHPosition* method
- *OE2DAtomDisplay.SetHCount* method
- *OEHydrogenPos* namespace

GetLabel

```
const std::string& GetLabel() const
```

Returns the string that is rendered as the atomic label at the coordinates returned by the *OE2DAtomDisplay.GetCoords* method. The color of the font is set by the

See also

- *OE2DAtomDisplay.SetLabel* method

GetLabelFont

```
const OEFont& GetLabelFont() const
```

Returns the font that is used when rendering the atomic label string of the *OE2DAtomDisplay* object.

See also

- *OE2DAtomDisplay.SetLabelFont* method
- *OE2DAtomDisplay.SetLabel* method
- *OEFont* class

GetMass

```
unsigned int GetMass() const
```

Returns the isotope information rendered as a super-script next to the atomic label. If zero, nothing is rendered. See example in *Figure: Example of isotope representation*.

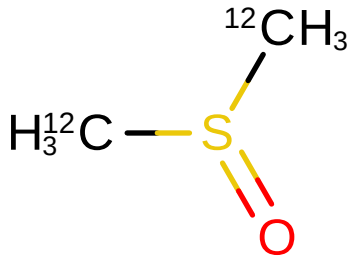


Fig. 5: **Example of isotope representation**

See also

- *OE2DAtomDisplay.SetMass* method

GetProperty

```
const std::string& GetProperty() const
```

Returns the string rendered next to the atom. If the string is empty, nothing is rendered.

See also

- *OE2DAtomDisplay.SetProperty* method
- *OE2DAtomDisplay.SetPropertyFont* method

GetPropertyFont

```
const OEFont& GetPropertyFont() const
```

Returns the font that is used when rendering the property string of the *OE2DAtomDisplay* object.

See also

- *OE2DAtomDisplay.SetPropertyFont* method
- *OE2DAtomDisplay.SetProperty* method
- *OEFont* class

GetPropertyOffset

```
const OE2DPoint& GetPropertyOffset() const
```

Returns the offset vector relative to the coordinates returned by *OE2DAtomDisplay.GetCoords* that is used to position the property label of the atom. These vectors are calculated during *OE2DMolDisplay* construction to minimize potentially clashing with the rest of the depiction. See *Figure: The property offset vector is relative to coordinates returned by the GetCoords method*.

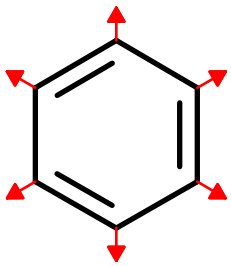


Fig. 6: **The property offset vector is relative to coordinates returned by the OE2DAtomDisplay::GetCoords method**

This allows for any string representable information to be rendered at a logical location near the atom. See example in *Example of indices as an atom property*.

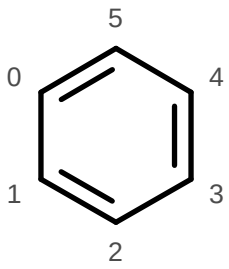


Fig. 7: **Example of indices as an atom property**

See also

- *OE2DAtomDisplay.SetPropertyOffset* method
- *OE2DPoint* class

GetRadical

```
unsigned int GetRadical() const
```

Returns the radical display type of the atom. The return value is taken from the *OERadicalDisplayType* namespace. See example in *Example of rendering radicals*.

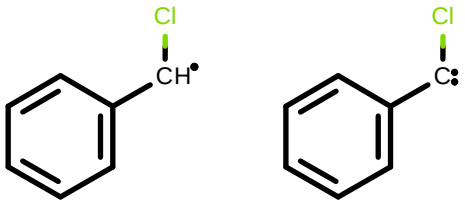


Fig. 8: **Example of rendering radicals**

See also

- *OE2DAtomDisplay.SetRadical* method
- *OERadicalDisplayType* namespace

HasLabel

```
bool HasLabel() const
```

Returns whether or not the atomic label string of the *OE2DAtomDisplay* object is empty.

See also

- *OE2DAtomDisplay.GetLabel* method
- *OE2DAtomDisplay.SetLabel* method

HasProperty

```
bool HasProperty() const
```

Returns whether or not the property string of the *OE2DAtomDisplay* object is empty.

See also

- *OE2DAtomDisplay.GetProperty* method
- *OE2DAtomDisplay.SetProperty* method

SetCharge

```
void SetCharge(int charge)
```

Sets the charge rendered next to the atomic label.

See also

- *OE2DAtomDisplay.GetCharge* method

SetCoords

```
void SetCoords(const OE2DPoint& coords)
```

Sets the exact coordinates of the atom in the rendered image.

See also

- *OE2DAtomDisplay.GetCoords* method
- *OE2DPoint* method

SetHCount

```
void SetHCount(unsigned int hcount)
```

Set the hydrogen count to render for this atom.

See also

- *OE2DAtomDisplay.GetHCount* method for more information on the meaning of various input integers.

SetHPosition

```
void SetHPosition(unsigned int hpos)
```

Sets the position of hydrogen label relative to the atomic label.

hpos
This value has to be from the *OEHydrogenPos* namespace.

See also

- *OE2DAtomDisplay.GetHPosition* method
- *OEHydrogenPos* namespace

SetLabel

```
void SetLabel(const std::string& label)
```

Sets the string that is rendered as the "atomic" label at the position returned by the *OE2DAtomDisplay.GetCoords* method.

See also

- *OE2DAtomDisplay.GetLabel* method

SetLabelFont

```
void SetLabelFont(const OEFont& font)
```

Sets the font that is used when rendering the atomic label of the *OE2DAtomDisplay* object.

See also

- *OE2DAtomDisplay.GetLabelFont* method

SetMass

```
void SetMass(unsigned int mass)
```

Sets the isotope information rendered next to the atomic label.

See also

- *OE2DAtomDisplay.GetMass* method

SetProperty

```
void SetProperty(const std::string& label)
```

Sets the string rendered next to the atom.

Warning

The preferred way to set the atom property strings is to use the *OE2DMolDisplayOptions.SetAtomPropertyFunctor* method. See example in the *Displaying Atom Properties* section.

See also

- *OE2DAtomDisplay.SetProperty* method
- *OE2DAtomDisplay.SetPropertyFont* method
- *OE2DMolDisplayOptions.SetAtomPropertyFunctor* method

SetPropertyFont

```
void SetPropertyFont(const OEFont& font)
```

Sets the font that is used when rendering the property string of the *OE2DAtomDisplay* object.

Warning

The preferred way to set the font of the atom property strings is to use the *OE2DMolDisplayOptions.SetAtomPropLabelFont* method.

See also

- *OE2DAtomDisplay.GetPropertyFont* method
- *OE2DAtomDisplay.GetProperty* method
- *OEFont* class
- *OE2DMolDisplayOptions.SetAtomPropLabelFont* method

SetPropertyOffset

```
void SetPropertyOffset(const OE2DPoint& offset)
```

Sets the offset vector that is used to position the property label of the atom.

See also

- *OE2DAtomDisplay.GetPropertyOffset* method
- *OE2DPoint* class

SetRadical

```
void  SetRadical(unsigned int radical)
```

Sets the radical type of the atom.

radical
This value has to be from the *OERadicalDisplayType* namespace.

See also

- *OE2DAtomDisplay.GetRadical* method
- *OERadicalDisplayType* namespace

4.1.2 OE2DBondDisplay

```
class OE2DBondDisplay : public OEBondDisplayBase
```

This class stores 2D depiction information of a bond. See *Figure: OEDepict TK bond display class hierarchy*.

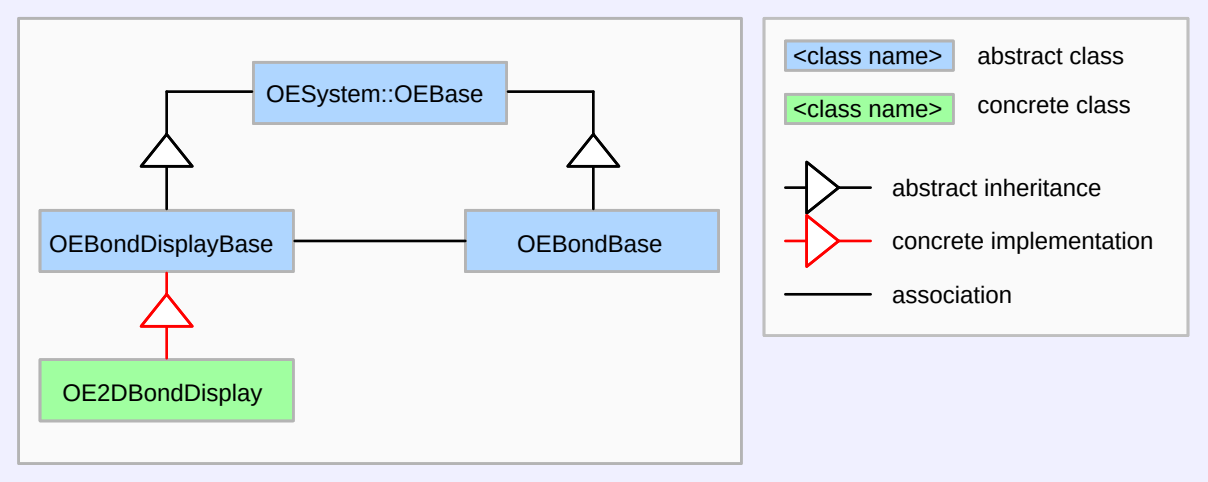


Fig. 9: **OEDepict TK bond display class hierarchy**

The following methods are publicly inherited from *OEBondDisplayBase*:

GetDataType	*IsDataType*	*SetVisible*
GetBond	*IsVisible*	

The *OE2DBondDisplay* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
pen used to draw the begin of the bond	*GetBgnPen*	*SetBgnPen*	*OEPen* class
pen used to draw the end of the bond	*GetEndPen*	*SetEndPen*	*OEPen* class
position of multi-line bonds	*GetDisplayPos*	*SetDisplayPos*	*OEBondDisplayPosition* namespace
rendered bond type	*GetDisplayType*	*SetDisplayType*	*OEBondDisplayType* namespace
rendered string as property	*GetProperty*	*SetProperty*	string
font of property string	*GetPropertyFont*	*SetPropertyFont*	*OEFont* class
position of property label	*GetPropertyOffset*	*SetPropertyOffset*	*OE2DPoint* class

Constructors

```
OE2DBondDisplay(const OEChem::OEBondBase* bond)
```

Construct an *OE2DBondDisplay* object that contains the style information for bond.

```
OE2DBondDisplay(const OE2DBondDisplay& rhs)
```

Copy constructor.

CreateCopy

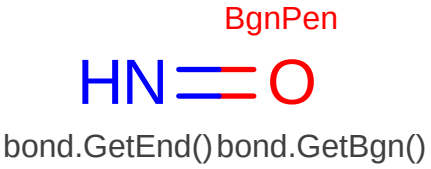
```
OESystem::OEBase* CreateCopy() const
```

Deep copy constructor that returns a pointer to a copy of the object. The memory returned is dynamically allocated and owned by the caller.

GetBgnPen

```
const OEPen& GetBgnPen() const
```

Returns the pen used to draw the bond from the “begin” atom coordinates to the middle of the bond.



See also

- *OE2DBondDisplay.SetBgnPen* method
- *OEPen* class

GetDisplayPos

```
unsigned int GetDisplayPos() const
```

Returns how multi-line bonds should be rendered. The return value is taken from the *OEBondDisplayPosition* namespace.

Note

The *OEBondDisplayPosition* namespace only affects bonds of the following types (as returned by *OE2DBondDisplay.GetDisplayType*):

- *OEBondDisplayType.Double*
- *OEBondDisplayType.AromaticDash*
- *OEBondDisplayType.SingleDouble*
- *OEBondDisplayType.SingleArom*
- *OEBondDisplayType.DoubleArom*
- *OEBondDisplayType.Aromatic*

See also

- *OE2DBondDisplay.SetDisplayPos* method
- *OEBondDisplayPosition* namespace

GetDisplayType

```
unsigned int GetDisplayType() const
```

Returns how bond orders, stereochemistry, and aromaticity is rendered for the bond. The return value is taken from the *OEBondDisplayType* namespace.

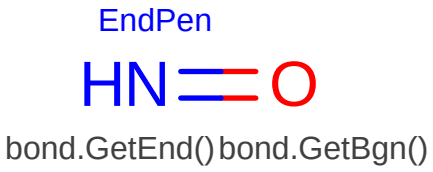
See also

- *OE2DBondDisplay.SetDisplayType* method
- *OEBondDisplayType* namespace

GetEndPen

```
const OEPen& GetEndPen() const
```

Returns the pen used to draw the bond from the "end" atom coordinates to the middle of the bond.



See also

- *OE2DBondDisplay.SetEndPen* method
- *OEPen* class

GetProperty

```
const std::string& GetProperty() const
```

Returns the string displayed next to the bond.

See also

- *OE2DBondDisplay.SetProperty* method
- *OE2DBondDisplay.SetPropertyFont* method

GetPropertyFont

```
const OEFont& GetPropertyFont() const
```

Returns the font that is used when displaying the property string of the *OE2DBondDisplay* object.

See also

- *OE2DBondDisplay.SetPropertyFont* method

- *OE2DBondDisplay.SetProperty* method
- *OEFont* class

GetPropertyOffset

```
const OE2DPoint& GetPropertyOffset() const
```

Returns the offset vector (relative to the middle of the bond) that is used to position the property label of the bond. See *Figure: The property offset vector is relative to the middle of the displayed bond*.

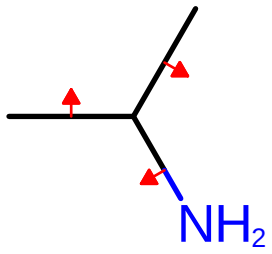


Fig. 10: **The property offset vector is relative to the middle of the displayed bond**

This allows for any string representable information to be rendered at a logical location near the bond. See example in *Example of displaying indices as bond property*.

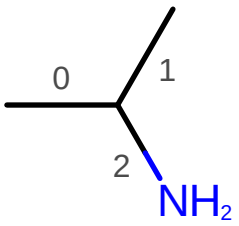


Fig. 11: **Example of displaying indices as bond property**

See also

- *OE2DBondDisplay.SetPropertyOffset* method
- *OE2DPoint* class

HasProperty

```
bool HasProperty() const
```

Return whether or not the property string of the *OE2DBondDisplay* object is empty.

See also

- *OE2DBondDisplay.GetProperty* method
- *OE2DBondDisplay.SetProperty* method

SetBgnPen

```
void SetBgnPen(const OEPen& pen)
```

Sets the pen used to draw the bond from the "begin" atom coordinates to the middle of the bond.

See also

- *OE2DBondDisplay.GetBgnPen* method
- *OEPen* class

SetDisplayPos

```
void SetDisplayPos(unsigned int pos)
```

Sets the display position of the bond.

pos
This value has to be from the *OEBondDisplayPosition* namespace.

See also

- *OE2DBondDisplay.GetDisplayPos* method
- *OEBondDisplayPosition* namespace

SetDisplayType

```
void SetDisplayType(unsigned int type)
```

Sets the display style of the bond.

type
This value has to be from the *OEBondDisplayType* namespace.

See also

- *OE2DBondDisplay.GetDisplayType* method
- *OEBondDisplayType* namespace

SetEndPen

```
void SetEndPen(const OEPen& pen)
```

Sets the pen used to draw the bond from the "end" atom coordinates to the middle of the bond.

See also

- *OE2DBondDisplay.GetEndPen* method
- *OEPen* class

SetProperty

```
void SetProperty(const std::string& label)
```

Sets the string displayed next to the bond.

Warning

The preferred way to set the bond property strings is to use *OE2DMolDisplayOptions.SetBondPropertyFunctor* method. See example in the *Displaying Bond Properties* section.

See also

- *OE2DBondDisplay.SetProperty* method
- *OE2DBondDisplay.SetPropertyFont* method
- *OE2DMolDisplayOptions.SetBondPropertyFunctor* method

SetPropertyFont

```
void SetPropertyFont(const OEFont& font)
```

Sets the font that is used when displaying the property string of the *OE2DBondDisplay* object.

Warning

The preferred way to set the font of the bond property strings is to use *OE2DMolDisplayOptions.SetBondPropLabelFont* method.

See also

- *OE2DBondDisplay.GetPropertyFont* method
- *OE2DBondDisplay.GetProperty* method
- *OEFont* class
- *OE2DMolDisplayOptions.SetBondPropLabelFont* method

SetPropertyOffset

```
void SetPropertyOffset(const OE2DPoint& offset)
```

Sets the offset vector (relative to the middle of the bond) that is used to position the property label of the bond.

See also

- *OE2DBondDisplay.GetPropertyOffset* method
- *OE2DPoint* class

4.1.3 OE2DMolDisplay

```
class OE2DMolDisplay : public OEMolDisplayBase
```

This class represents *OE2DMolDisplay* that stores the depiction information (such as atom coordinates, molecule scaling, representation styles etc.) of a molecule. See *Figure: OEDepict TK molecule display class hierarchy*.

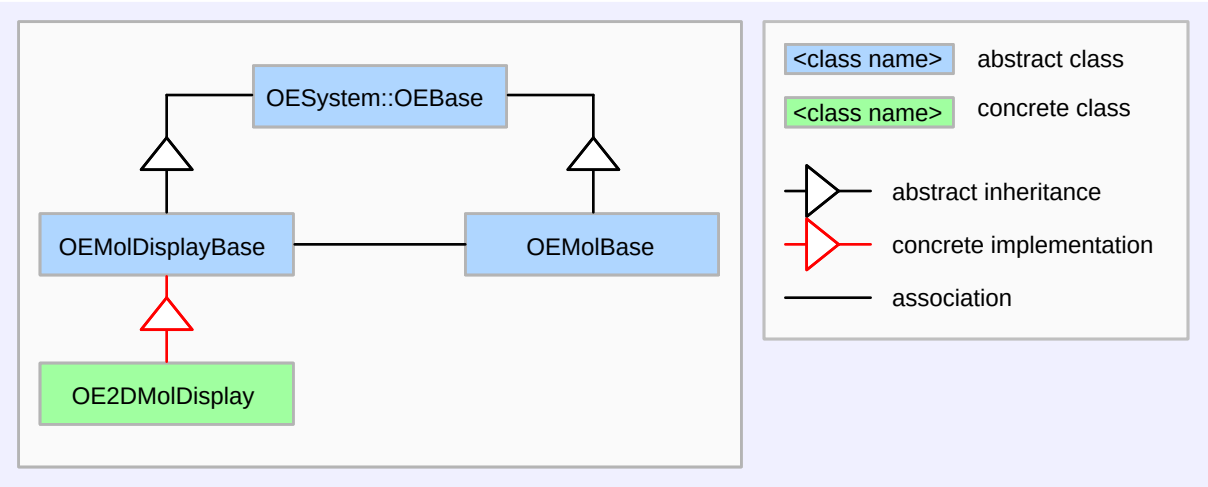


Fig. 12: **OEDepict TK molecule display class hierarchy**

See also

- *OE2DMolDisplayOptions* class
- *OE2DAtomDisplay* class
- *OE2DBondDisplay* class

The following methods are publicly inherited from *OEMolDisplayBase*:

GetDataType	*GetMolecule*	*IsDataType*

Constructors

```
OE2DMolDisplay(const OEChem::OEMolBase& mol)
```

Initializes an *OE2DMolDisplay* object using default display options.

mol
The molecule for which display information is stored in the *OE2DMolDisplay* object.

See also

- *OE2DMolDisplayOptions.Constructors* method for the list of default display options
- *OE2DMolDisplay.IsValid* method

```
OE2DMolDisplay(const OEChem::OEMolBase& mol,
               const OE2DMolDisplayOptions& opts)
```

Initializes an *OE2DMolDisplay* object using the given display options.

mol
The molecule for which display information is stored in the *OE2DMolDisplay* object.

opts
The *OE2DMolDisplayOptions* object that stores properties that determine the styles of the molecule depiction.

Warning

- An *OE2DMolDisplay* object does not make a copy of the OEMolBase object from which it is initialized but only stores its pointer. Therefore, the user responsibility is to not let the molecule go out of scope, before the corresponding *OE2DMolDisplay* object.
- Changing the OEMolBase object from which *OE2DMolDisplay* object is initialized has no effect on the display itself after initialization. However, the *OE2DMolDisplay* object become invalid if the molecule graph is modified (atoms or bonds added or deleted).

Note

- A warning is thrown when an *OE2DMolDisplay* object is initialized with a molecule with 3D coordinates.

- An *OE2DMolDisplay* object is considered invalid and an error is thrown when it is initialized:
  - with an empty molecule (*i.e.* molecule with no atoms)
  - with a molecule that has neither 2D nor 3D coordinates

  See also: *OE2DMolDisplay.IsValid* method

```
OE2DMolDisplay(const OE2DMolDisplay& rhs)
```

Copy constructor.

CreateCopy

```
OESystem::OEBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OE2DMolDisplay* object is dynamically allocated and owned by the caller.

GetAtomDisplay

```
OE2DAtomDisplay* GetAtomDisplay(const OEChem::OEAtomBase* atom)
```

Returns the pointer of the *OE2DAtomDisplay* object that stores the depiction information of the given atom.

```
const OE2DAtomDisplay* GetAtomDisplay(const OEChem::OEAtomBase* atom) const
```

Returns the const pointer of the *OE2DAtomDisplay* object that stores the depiction information of the given atom.

Note

The *OE2DMolDisplay.GetAtomDisplay* methods returns a zero pointer if the given atom does not correspond to any display atom stored in the *OE2DMolDisplay* object. It is a good programming practice to check the returned pointer before using it. See code snippet below.

```
OE2DAtomDisplay adisp = disp.GetAtomDisplay(atom);
if (adisp != null && adisp.IsVisible()) {
    // do something
}
```

See also

- *OE2DAtomDisplay* class

GetAtomDisplays

```
OESystem::OEIterBase<OE2DAtomDisplay>* GetAtomDisplays() const
```

Returns an iterator over all *OE2DAtomDisplay* objects stored in the *OE2DMolDisplay* object.

```
OESystem::OEIterBase<OE2DAtomDisplay>* GetAtomDisplays(const OESystem::OEUnaryPredicate
↪<OEChem::OEAtomBase>& pred) const
```

Returns an iterator over only those *OE2DAtomDisplay* objects for which the corresponding atom passes the given predicate. For example, the following code snippet shows how to access the atom displays of oxygen atoms:

```
for (OE2DAtomDisplay ad : disp.GetAtomDisplays(new OEHasAtomicNum(OEElemNo.O))) {
    if (ad.IsVisible()) {
        // do something
    }
}
```

See also

- *OE2DAtomDisplay* class
- `Listing 4` example in the *Customizing Molecule Depiction* section

GetBondDisplay

```
OE2DBondDisplay* GetBondDisplay(const OEChem::OEBondBase* bond)
```

Returns the pointer of the *OE2DBondDisplay* object, that stores the depiction information of the given bond.

```
const OE2DBondDisplay* GetBondDisplay(const OEChem::OEBondBase* bond) const
```

Returns the const pointer of the *OE2DBondDisplay* object, that stores the depiction information of the given bond.

See also

- *OE2DBondDisplay* class

Note

The `OE2DMolDisplay.GetBondDisplay` methods returns a zero pointer if the given bond does not correspond to any display bond stored in the *OE2DMolDisplay* object. It is a good programming practice to check the returned pointer before using it. See code snippet below.

```
OE2DBondDisplay bdisp = disp.GetBondDisplay(bond);
if (bdisp != null && bdisp.IsVisible()) {
    // do something
}
```

See also

- *OE2DBondDisplay* class

GetBondDisplays

```
OESystem::OEIterBase<OE2DBondDisplay>* GetBondDisplays() const
```

Returns an iterator over all *OE2DBondDisplay* objects stored in the *OE2DMolDisplay* object.

```
OESystem::OEIterBase<OE2DBondDisplay>* GetBondDisplays(const OESystem::OEUnaryPredicate
↪<OEChem::OEBondBase>& pred) const
```

See also

- *OE2DBondDisplay* class

Returns an iterator over only those *OE2DBondDisplay* objects for which the corresponding bond passes the given predicate. For example, the following code snippet shows how to access the bond displays of aromatic bonds:

```
for (OE2DBondDisplay bd : disp.GetBondDisplays(new OEIsAromaticBond())) {
    if (bd.IsVisible()) {
        // do something
    }
}
```

See also

- *OE2DBondDisplay* class
- `Listing 4` example in the *Customizing Molecule Depiction* section

GetDataType

```
const void* GetDataType() const
```

This function is used to perform run-time type identification.

See also

- `OEBase.GetDataType` method in the **OEChem TK** manual

GetHeight

```
double GetHeight() const
```

Returns the height of the displayed molecule.

See also

- *Controlling the Size of Depicted Molecules* section
- `OE2DMolDisplay.GetScale` method

- *OE2DMolDisplay.GetWidth* method

GetLayer

```
OEImage& GetLayer(unsigned int position, unsigned int idx = 0)
const OEImage& GetLayer(unsigned int position, unsigned int idx = 0) const
```

Provides access to the layers of the *OE2DMolDisplay* object. Currently, an *OE2DMolDisplay* object has two layers (one *OELayerPosition.Above* and one *OELayerPosition.Below*) that allows drawing of objects over or underneath the molecular structure, respectively.

position
: This value has to be from the *OELayerPosition* namespace.

idx
: This parameter is currently not used.

See also

- *OEImage* class
- *OELayerPosition* namespace

GetOptions

```
const OE2DMolDisplayOptions& GetOptions() const
```

Returns the options used to initialized the *OE2DMolDisplay* object.

See also

- *OE2DMolDisplayOptions* class

GetScale

```
double GetScale() const
```

Returns the scaling factor of the displayed molecule.

See also

- *Controlling the Size of Depicted Molecules* section
- *OE2DMolDisplay.GetHeight* method
- *OE2DMolDisplay.GetWidth* method

GetWidth

```
double GetWidth() const
```

Returns the width of the displayed molecule.

See also

- *Controlling the Size of Depicted Molecules* section
- *OE2DMolDisplay.GetHeight* method
- *OE2DMolDisplay.GetScale* method

IsDataType

```
bool IsDataType(const void *) const
```

Returns whether type is the same as the instance this method is called on.

See also

- OEBase.IsDataType method in the **OEChem TK** manual

IsValid

```
bool IsValid() const
```

Returns whether the *OE2DMolDisplay* object was initialized successfully. If initialization was attempted with an empty molecule or a molecule with no coordinates (neither 2D nor 3D), then this method returns false. An *OE2DMolDisplay* object becomes invalid if the molecule, from which it was initialized, is modified. For example, atom or bonds added to or deleted from the molecule.

Hint

It is highly recommend to check whether a molecule display is valid after initialization.

4.1.4 OE2DMolDisplayOptions

```
class OE2DMolDisplayOptions
```

This class represents the *OE2DMolDisplayOptions* class that encapsulates properties that determine how a molecule is depicted.

The *OE2DMolDisplayOptions* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
aromatic style	*GetAromaticStyle*	*SetAromaticStyle*	*OEAromaticStyle*
atomic number colors	*GetAtomColor*	*SetAtomColor*	OEColor
atom color style	*GetAtomColorStyle*	*SetAtomColorStyle*	*OEAtomColorStyle*
atom label font	*GetAtomLabelFont*	*SetAtomLabelFont*	*OEFont*
atom label font scale	*GetAtomLabelFontSca*	*SetAtomLabelFontSca*	positive floating point number in the range of `[0.5, 3.0]` or `0.0`
atom property font	*GetAtomPropLabelFor*	*SetAtomPropLabelFor*	*OEFont*
atom property label font scale	*GetAtomPropLabelFor*	*SetAtomPropLabelFor*	positive floating point number in the range of `[0.5, 2.0]`
atom property initializer	*GetAtomPropertyFunc*	*SetAtomPropertyFunc*	*OEDisplayAtomPropBase*
atom stereo style	*GetAtomStereoStyle*	*SetAtomStereoStyle*	*OEAtomStereoStyle*
atom SVG markup	*GetAtomSVGMarkupFur*	*SetAtomSVGMarkupFur*	*OEAtomSVGMarkupBase*
atom visibility initializer	*GetAtomVisibilityFu*	*SetAtomVisibilityFu*	`OEUnaryAtomPred`
background color	*GetBackgroundColor*	*SetBackgroundColor*	OEColor
bond color style	*GetBondColorStyle*	*SetBondColorStyle*	*OEBondColorStyle*
bond line gap scale	*GetBondLineGapScale*	*SetBondLineGapScale*	positive floating point number in the range of `[0.5, 2.0]`
bond line atom label gap scale	*GetBondLineAtomLabe*	*SetBondLineAtomLabe*	positive floating point number in the range of `[0.5, 2.0]`
bond property font	*GetBondPropLabelFor*	*SetBondPropLabelFor*	*OEFont*
bond property label font scale	*GetBondPropLabelFor*	*SetBondPropLabelFor*	positive floating point number in the range of `[0.5, 2.0]`
bond property initializer	*GetBondPropertyFunc*	*SetBondPropertyFunc*	*OEDisplayBondPropBase*
bond stereo style	*GetBondStereoStyle*	*SetBondStereoStyle*	*OEBondStereoStyle*
bond pen	*GetDefaultBondPen*	*SetDefaultBondPen*	*OEPen*
bond width scaling	*GetBondWidthScaling*	*SetBondWidthScaling*	boolean
bond SVG markup	*GetBondSVGMarkupFur*	*SetBondSVGMarkupFur*	*OEBondSVGMarkupBase*
explicit atom label angle	*GetExplicitAtomLabe*	*SetExplicitAtomLabe*	positive floating point number in the range of `[120.0, 180.0]`
height	*GetHeight*	*SetHeight*	positive floating point number
hydrogen style	*GetHydrogenStyle*	*SetHydrogenStyle*	*OEHydrogenStyle*
margin(s)	*GetMargin*	*SetMargin*, *SetMargins*	positive floating point number in the range of `[2.5, 25.0]`
scale	*GetScale*	*SetScale*	positive floating point number
super atom label font	*GetSuperAtomLabelFc*	*SetSuperAtomLabelFc*	*OEFont*
super atom display style	*GetSuperAtomStyle*	*SetSuperAtomStyle*	*OESuperAtomStyle*
protective group label font	*GetProtectiveGroupL*	*SetProtectiveGroupL*	*OEFont*
protective group display style	*HasProtectiveGroupS*	*SetProtectiveGroupS*	*OEProtectiveGroupStyle*
title font	*GetTitleFont*	*SetTitleFont*	*OEFont*

continues on next page

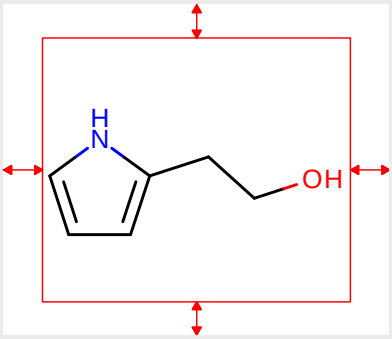
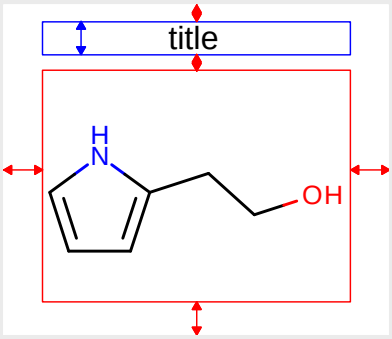
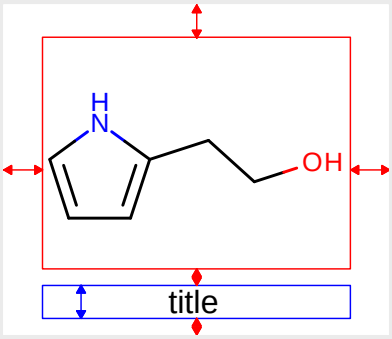
Table 1 – continued from previous page

Property	Get method	Set method	Corresponding namespace / class / type
title font scale	*GetTitleFontScale*	*SetTitleFontScale*	positive floating point number in the range of `[0.5, 2.0]` or `0.0`
title height	*GetTitleHeight*	*SetTitleHeight*	*OETitleLocation*
title location	*GetTitleLocation*	*SetTitleLocation*	*OETitleLocation*
width	*GetWidth*	*SetWidth*	positive floating point number

The images below illustrate the three possible molecule display layouts depending on the position of the title. These layouts can be controlled by the following parameters:

- The width and the height can be set either by the constructor of the *OE2DMolDisplayOptions* class, by calling the *OE2DMolDisplayOptions.SetDimensions* method or by the *SetWidth* and *SetHeight* methods, respectively.
- The location of the title area (blue box) can be set by the *SetTitleLocation* method. The height of the title area (blue arrows) can be modified by invoking the *SetTitleHeight* method.
- The padding around the molecule (red arrows) can be changed by calling either the *SetMargin* or the *SetMargins* methods.

Table 2: **Example of molecule display layouts depending on position of the title**

title hidden	title at the top	title at the bottom
		

See also

- *OE2DMolDisplay* class
- *Controlling the Size of Depicted Molecules* section

Hint

Even though **OEDepict TK** provides access to manipulate the properties of the atom and bond displays after the *OE2DMolDisplay* object is constructed, it is highly recommended to determine the style of the depiction by using the *OE2DMolDisplayOptions* class. Only by knowing all properties (such as labels, font styles and sizes *etc.*) in advance can ensure the **best depiction layout** *i.e.* that the molecule diagram is rendered without any label clippings and the labels are displayed with the minimum number of overlaps.

Constructors

```
OE2DMolDisplayOptions()
```

Default constructor that initializes an *OE2DMolDisplayOptions* object with the following properties:

See example in *Molecule depiction with default options*)

Table 3: **Default parameters of OE2DMolDisplayOptions**

Property	Default value
aromatic style	*OEAromaticStyle.Default*
atomic number colors	initialized based on *OEAtomColorStyle.Default*
atom color style	*OEAtomColorStyle.Default*
atom label font	*OEDepict.OEDefaultFont*
atom label font scale	1.0
atom property font	OEFont(OEColor(75, 75, 75))
atom property label font scale	1.0
atom property initializer	*OEDisplayNoAtomProp*
atom stereo style	*OEAtomStereoStyle.Default*
atom SVG markup	*OEAtomSVGNoMarkup*
atom visibility initializer	`OEIsTrueAtom`
background color	*OEDepict.OELightBackgroundColor* initialized based on *OEAtomColorStyle.Default*
bond color style	*OEBondColorStyle.Default*
bond line gap scale	1.0
bond line atom label gap scale	1.0
bond property font	OEFont(OEColor(75, 75, 75))
bond property label font scale	1.0
bond property initializer	*OEDisplayNoBondProp*
bond stereo style	*OEBondStereoStyle.Default*
bond pen	*OEDepict.OEBlackPen*
bond width scaling	*false*
bond SVG markup	*OEBondSVGNoMarkup*
explicit atom label angle	170.0 (degree)
height	0.0 (initialized based on scale)
hydrogen style	*OEHydrogenStyle.Default*
margins	left and right margin *5.0%* of the width; top and bottom margin *5.0%* of the height of the molecule display
scale	*OEScale.AutoScale*
super atom label font	OEFont(OEPink)
super atom display style	*OESuperAtomStyle.Default*
protective group label font	OEFont(OEPink)
protective group display style	*OEProtectiveGroupStyle.Default*
title font	*OEDepict.OEDefaultFont*
title font scale	1.0
title location	*OETitleLocation.Default*
title height	*10%* of the height of the molecule display
width	0.0 (initialized based on scale)

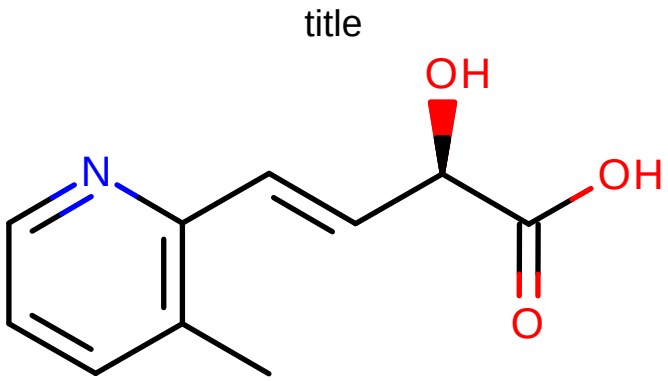


Fig. 13: **Molecule depiction with default options**

```
OE2DMolDisplayOptions(double scale)
```

Creates an *OE2DMolDisplayOptions* object with the specified scaling.

scale
A non-negative number that controls the magnification of the depicted molecule.

```
OE2DMolDisplayOptions(double width, double height, double scale)
```

Creates an *OE2DMolDisplayOptions* object with the specified dimension and scaling.

```
OE2DMolDisplayOptions(const OE2DMolDisplayOptions& rhs)
```

Copy constructor.

GetAromaticStyle

```
unsigned int GetAromaticStyle() const
```

Returns the style that controls how aromatic rings are displayed. The return value is taken from the *OEAromaticStyle* namespace.

See also

- *OE2DMolDisplayOptions.SetAromaticStyle* method
- *OEAromaticStyle* namespace

GetAtomSVGMarkupFunctor

```
const OEAtomSVGMarkupBase& GetAtomSVGMarkupFunctor() const
```

Returns the functor that defines how atoms are marked in `svg` image. By default, residues are not marked (*OEAtomSVGNoMarkup*).

See also

- *OE2DMolDisplayOptions.SetAtomSVGMarkupFunctor* method

GetAtomColor

```
const OESystem::OEColor& GetAtomColor(unsigned int atomic) const
```

Returns the color associated with a specific atomic number.

See also

- *OE2DMolDisplayOptions.SetAtomColor* method
- OEColor class
- *Appendix: Element coloring (CPK)* section

GetAtomColorStyle

```
unsigned int GetAtomColorStyle() const
```

Returns the style that is used to define the color of the background and the default color of atom labels. The return value is taken from the *OEAtomColorStyle* namespace.

See also

- *OE2DMolDisplayOptions.SetAtomColorStyle* method
- *OEAtomColorStyle* namespace

GetAtomLabelFont

```
const OEFont& GetAtomLabelFont() const
```

Returns the font that is used to initialize the depiction style of the atom labels (*OE2DAtomDisplay.GetLabelFont*).

See also

- *OE2DMolDisplayOptions.SetAtomLabelFont* method
- *OE2DMolDisplayOptions.SetAtomLabelFontScale* method
- *OEFont* class

GetAtomLabelFontScale

```
double GetAtomLabelFontScale() const
```

Returns the multiplier that can be used to increase or decrease the size of the atom label fonts.

See also

- *OE2DMolDisplayOptions.SetAtomLabelFont* method
- *OE2DMolDisplayOptions.SetAtomLabelFontScale* method

GetAtomPropLabelFont

```
const OEFont& GetAtomPropLabelFont() const
```

Returns the font that is used to initialize the depiction style of the atom property labels (*OE2DAtomDisplay.GetPropertyFont*).

See also

- *OE2DMolDisplayOptions.SetAtomPropLabelFont* method
- *OEFont* class

GetAtomPropLabelFontScale

```
double GetAtomPropLabelFontScale() const
```

Returns the multiplier that can be used to increase or decrease the size of the atom property label fonts.

See also

- *OE2DMolDisplayOptions.SetAtomPropLabelFont* method
- *OE2DMolDisplayOptions.SetAtomPropLabelFontScale* method

GetAtomPropertyFunctor

```
const OEDisplayAtomPropBase& GetAtomPropertyFunctor() const
```

Returns the functor that is used to initialize the atom property labels (*OE2DAtomDisplay.GetProperty*).

See also

- *OE2DMolDisplayOptions.SetAtomPropertyFunctor* method
- *OEDisplayAtomPropBase* class
- *OEDisplayAtomIdx* class
- *OEDisplayAtomMapIdx* class
- *OEDisplayNoAtomProp* class

GetAtomVisibilityFunctor

```
const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& GetAtomVisibilityFunctor() const
```

Returns the functor that is used to identify visible atoms. The default functor is the `OEIsTrueAtom` class which results in all atoms being displayed.

See also

- *OE2DMolDisplayOptions.SetAtomVisibilityFunctor* method
- OEUnaryAtomPred

GetAtomStereoStyle

```
unsigned int GetAtomStereoStyle() const
```

Returns the style that controls what atom stereo information is displayed. The return value is taken from the *OEAtomStereoStyle* namespace.

See also

- *OE2DMolDisplayOptions.SetAtomStereoStyle* method
- *OEAtomStereoStyle* namespace

GetBackgroundColor

```
const OESystem::OEColor& GetBackgroundColor() const
```

Returns the color that is used to clear the background of an image (by calling *OEImageBase.Clear* method) before rendering the molecule.

See also

- *OE2DMolDisplayOptions.SetBackgroundColor* method
- OEColor class
- *OERenderMolecule* function

GetBondColorStyle

```
unsigned int GetBondColorStyle() const
```

Returns the style that controls how bonds are colored. The return value is taken from the *OEBondColorStyle* namespace.

See also

- *OE2DMolDisplayOptions.SetBondColorStyle* method

- *OEBondColorStyle* namespace

GetBondLineGapScale

```
double GetBondLineGapScale() const
```

Returns the multiplier that can be used to increase or decrease the gap between the lines of double and triple bonds.

See also

- *OE2DMolDisplayOptions.SetBondLineGapScale* method

GetBondLineAtomLabelGapScale

```
double GetBondLineAtomLabelGapScale() const
```

Returns the multiplier that can be used to increase or decrease the gap between the line(s) of bonds and the adjacent atom labels.

See also

- *OE2DMolDisplayOptions.SetBondLineAtomLabelGapScale* method

GetBondPropLabelFont

```
const OEFont& GetBondPropLabelFont() const
```

Returns the font that is used to initialize the depiction style of the bond property labels (*OE2DBondDisplay.GetPropertyFont*).

See also

- *OE2DMolDisplayOptions.SetBondPropLabelFont* method
- *OEFont* class

GetBondPropLabelFontScale

```
double GetBondPropLabelFontScale() const
```

Returns the multiplier that can be used to increase or decrease the size of the bond property label fonts.

See also

- *OE2DMolDisplayOptions.SetBondPropLabelFont* method
- *OE2DMolDisplayOptions.SetBondPropLabelFontScale* method

GetBondPropertyFunctor

```
const OEDisplayBondPropBase& GetBondPropertyFunctor() const
```

Returns the functor that is used to initialize the bond property labels (*OE2DBondDisplay.GetProperty*).

See also

- *OE2DMolDisplayOptions.SetBondPropertyFunctor* method
- *OEDisplayBondPropBase* class
- *OEDisplayBondIdx* class
- *OEDisplayNoBondProp* class

GetBondStereoStyle

```
unsigned int GetBondStereoStyle() const
```

Returns the style that controls what bond stereo information is displayed. The return value is taken from the *OEBondStereoStyle* namespace.

See also

- *OE2DMolDisplayOptions.SetBondStereoStyle* method
- *OEBondStereoStyle* namespace

GetBondSVGMarkupFunctor

```
const OEBondSVGMarkupBase& GetBondSVGMarkupFunctor() const
```

Returns the functor that defines how bonds are marked in svg image. By default, residues are not marked (*OEBondSVGNoMarkup*).

See also

- *OE2DMolDisplayOptions.SetBondSVGMarkupFunctor* method

GetBondWidthScaling

```
bool GetBondWidthScaling() const
```

Returns whether the line width of the bond are increase or decreased based on the molecule scaling factor (*OE2DMolDisplayOptions.GetScale*)

See also

- *OE2DMolDisplayOptions.SetBondWidthScaling* method

GetDefaultBondPen

```
const OEPen& GetDefaultBondPen() const
```

Returns the pen that is used to initialize the bond pens (*OE2DBondDisplay.GetBgnPen* and *OE2DBondDisplay.GetBgnPen*).

See also

- *OE2DMolDisplayOptions.SetDefaultBondPen* method
- *OEPen* class

GetExplicitAtomLabelAngle

```
double GetExplicitAtomLabelAngle() const
```

In the case of an almost linear X-C-X bond, an explicit "C" label will be drawn if the angle is more that the returned value of this function. 180.0 degree means that the "C" label will never be drawn.

See also

- *OE2DMolDisplayOptions.SetExplicitAtomLabelAngle* method

GetHeight

```
double GetHeight() const
```

Returns the vertical limit into which the molecule has to be fitted.

See also

- *OE2DMolDisplayOptions.SetDimensions* method
- *OE2DMolDisplayOptions.SetHeight* method

GetHydrogenStyle

```
unsigned int GetHydrogenStyle() const
```

Returns the style that controls of how implicit and explicit hydrogens are displayed. The return value is taken from the *OEHydrogenStyle* namespace.

See also

- *OE2DMolDisplayOptions.SetHydrogenStyle* method
- *OEHydrogenStyle* namespace

GetMargin

```
double GetMargin(unsigned int margin) const
```

Returns the ratio of a specific margin of the *OE2DMolDisplayOptions* object.

margin
: This value has to be from the *OEMargin* namespace.

See also

- *OE2DMolDisplayOptions.SetMargin* method
- *OE2DMolDisplayOptions.SetMargins* method
- *OEMargin* namespace

GetProtectiveGroupLabelFont

```
const OEFont& GetProtectiveGroupLabelFont() const
```

Returns the font that is used to display the protective group labels.

See also

- *OE2DMolDisplayOptions.SetProtectiveGroupLabelFont* method
- *OE2DMolDisplayOptions.SetProtectiveGroupStyle* method
- *OE2DMolDisplayOptions.SetSuperAtomStyle* method
- *OEFont* class

GetProtectiveGroupStyle

```
unsigned int GetProtectiveGroupStyle() const
```

Returns the style that controls whether or not specific pre-defined protective group are contracted and labeled with corresponding abbreviations. The return value is taken from the *OEProtectiveGroupStyle* namespace.

See also

- *OE2DMolDisplayOptions.HasProtectiveGroupStyle* method
- *OE2DMolDisplayOptions.SetProtectiveGroupStyle* method
- *OE2DMolDisplayOptions.SetProtectiveGroupLabelFont* method
- *OE2DMolDisplayOptions.SetSuperAtomStyle* method
- *OESuperAtomStyle* namespace

Warning

This is a deprecated API. Please use *OE2DMolDisplayOptions.HasProtectiveGroupStyle* to determine which protective groups are set.

GetScale

```
double GetScale() const
```

Returns the scaling factor used to magnify a molecule.

See also

- *OE2DMolDisplayOptions.SetDimensions* method
- *OE2DMolDisplayOptions.SetScale* method

GetSuperAtomLabelFont

```
const OEFont& GetSuperAtomLabelFont() const
```

Returns the font that is used to display the abbreviations of contracted functional groups.

See also

- *OE2DMolDisplayOptions.SetSuperAtomLabelFont* method
- *OE2DMolDisplayOptions.SetSuperAtomStyle* method
- *OE2DMolDisplayOptions.SetProtectiveGroupStyle* method
- *OEFont* class

GetSuperAtomStyle

```
unsigned int GetSuperAtomStyle() const
```

Returns the style that controls whether or not specific pre-defined functional groups are contracted and labeled with corresponding abbreviations. The return value is taken from the *OESuperAtomStyle* namespace.

See also

- *OE2DMolDisplayOptions.SetSuperAtomStyle* method
- *OE2DMolDisplayOptions.SetSuperAtomLabelFont* method
- *OE2DMolDisplayOptions.SetProtectiveGroupStyle* method
- *OESuperAtomStyle* namespace

GetTitleFont

Returns the font that is used to display the title of the molecule (*i.e.* the string returned by the `OEMolBase.GetTitle` method).

```
const OEFont& GetTitleFont() const
```

See also

- *OE2DMolDisplayOptions.SetTitleFont* method
- *OE2DMolDisplayOptions.SetTitleFontScale* method
- *OEFont* class

GetTitleFontScale

Returns the multiplier that can be used to increase or decrease the size of the title font.

```
double GetTitleFontScale() const
```

See also

- *OE2DMolDisplayOptions.SetTitleFontScale* method
- *OE2DMolDisplayOptions.SetTitleFont* method

GetTitleLocation

```
unsigned int GetTitleLocation() const
```

Returns the position of the molecule title. The return value is taken from the *OETitleLocation* namespace.

See also

- *OE2DMolDisplayOptions.SetTitleLocation* method
- *OETitleLocation* namespace

GetTitleHeight

```
double GetTitleHeight() const
```

Returns the height of the title area.

See also

- *OE2DMolDisplayOptions.SetTitleHeight* method

GetWidth

```
double GetWidth() const
```

Returns the horizontal limit into which the molecule has to be fitted.

See also

- *OE2DMolDisplayOptions.SetDimensions* method
- *OE2DMolDisplayOptions.SetWidth* method

HasProtectiveGroupStyle

```
bool HasProtectiveGroupStyle(unsigned int style) const
```

Returns true if the specific pre-defined protective group style is contracted and labeled with corresponding abbreviations.

style
: This value has to be from the *OEProtectiveGroupStyle* namespace.

SetAromaticStyle

```
void SetAromaticStyle(unsigned int style)
```

Sets the style that controls how aromatic rings are displayed.

style
: This value has to be from the *OEAromaticStyle* namespace.

Example: (*Figure: Example of using the SetAromaticStyle method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetAromaticStyle(OEAromaticStyle.Circle);
```

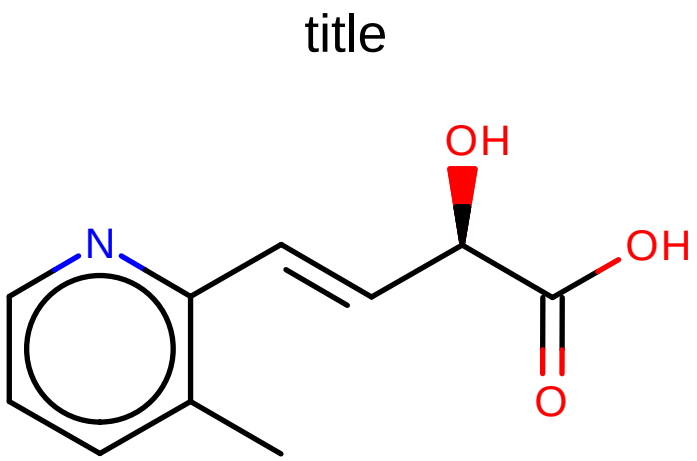


Fig. 14: **Example of using the SetAromaticStyle method**

See also

- *OE2DMolDisplayOptions.GetAromaticStyle* method
- *OEAromaticStyle* namespace

SetAtomColor

```
void SetAtomColor(unsigned int atomic,
                  const OESystem::OEColor& color)
```

Sets the color associated with a specific atomic number.

atomic
This value has to be from the `OEElemNo` namespace of **OEChem TK**.

color
The color being associated with the given atomic number.

Example: (*Figure: Example of using the SetAtomColor method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetAtomColor(OEElemNo.O, new OEColor(80, 0, 0)); // very dark red
opts.SetAtomColor(OEElemNo.N, new OEColor(0, 0, 80)); // very dark blue
```

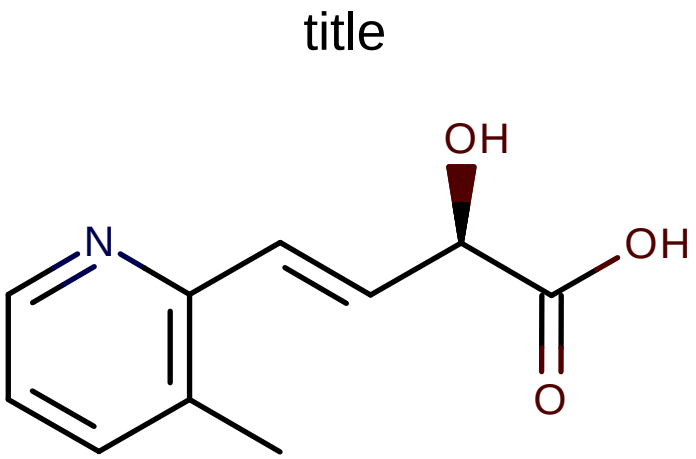


Fig. 15: **Example of using the SetAtomColor method**

See also

- *OE2DMolDisplayOptions.GetAtomColor* method
- OEColor class
- *Appendix: Element coloring (CPK)* section

Note

- The color of the atom also depends on the atom color style (*OE2DMolDisplayOptions.GetAtomColorStyle*).
- Setting atom color style (*OE2DMolDisplayOptions.SetAtomColorStyle*) to WhiteMonochrome sets white background and black color for all atoms. Likewise, setting atom color style to BlackMonochrome sets black background and white color for all atoms. But, user can choose to have another color for all atoms (in monochrome atom color style) by setting another atom color (*OE2DMolDisplayOptions.SetAtomColor*) for Carbon atoms. The example in *Listing 4* demonstrates setting non-default atom colors in monochrome mode. Setting atom color for other than Carbon atoms sets color of only specified atom types, not for all atoms.

SetAtomColorStyle

```
void SetAtomColorStyle(unsigned int style)
```

Sets the style that is used to define the color of the background and the default color of atom labels.

style
This value has to be from the *OEAtomColorStyle* namespace.

Example: (*Figure: Example of using the SetAtomColorStyle method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetAtomColorStyle(OEAtomColorStyle.BlackMonochrome);
```

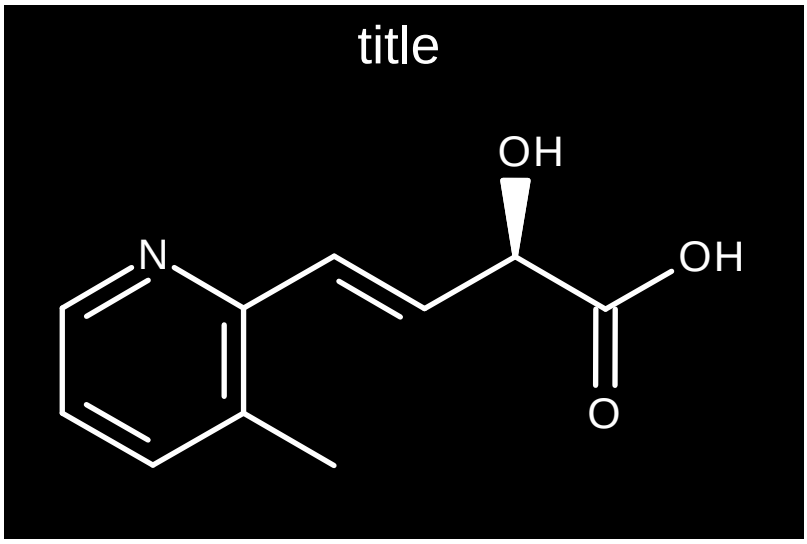


Fig. 16: **Example of using the SetAtomColorStyle method**

See also

- *OE2DMolDisplayOptions.GetAtomColorStyle* method
- *OEAtomColorStyle* namespace

SetAtomLabelFont

```
void SetAtomLabelFont(const OEFont& font)
```

Sets the font that is used to initialize the depiction style of the atom labels (*OE2DAtomDisplay.GetLabelFont*).

Example: (*Figure: Example of using the SetAtomLabelFont method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
OEFont font = new OEFont();
font.SetFamily(OEFontFamily.Courier);
font.SetStyle(OEFontStyle.Bold);
opts.SetAtomLabelFont(font);
```

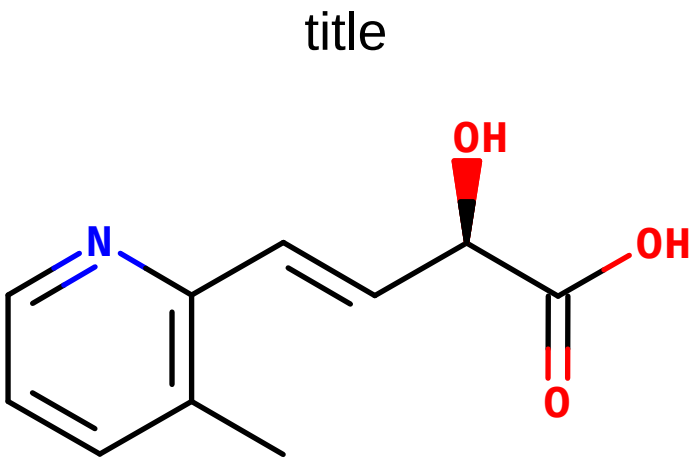


Fig. 17: **Example of using the SetAtomLabelFont method**

Note

- The color of the atom label fonts also depends on the atom color style (*OE2DMolDisplayOptions.GetAtomColorStyle*).
- The size of fonts of the atom labels also depends on:
  - the scaling factor used to fit a molecule to a given dimension (*OE2DMolDisplayOptions.GetScale*)
  - the multiplier set by the *OE2DMolDisplayOptions.SetAtomLabelFontScale* method

  See example in *Figure: Example of scaling the font of the atom label along with the molecule*

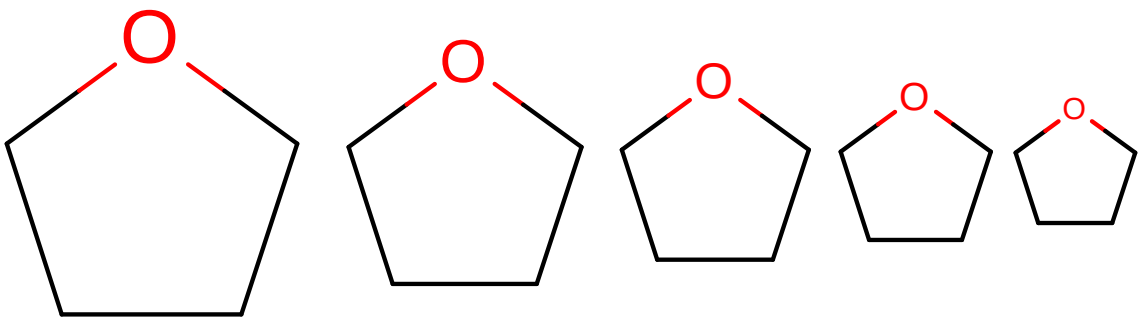


Fig. 18: **Example of scaling the font of the atom label along with the molecule**

See also

- *OE2DMolDisplayOptions.GetAtomLabelFont* method

- *OEFont* class

SetAtomLabelFontScale

```
void SetAtomLabelFontScale(double scale)
```

Sets the multiplier that can be used increase or decrease the size of the fonts of the atom labels.

scale

This value has to be either `0.0` or in a range of `[0.5, 3.0]`. See examples in *Figure: Examples of scaling the font of the atom label relative to the molecule*

In case of `0.0`, the atom labels are not scaled with the molecule but rather fixed font size are used regardless of the size of the molecule. See example in *Figure: Example of atom labels with fixed font size*

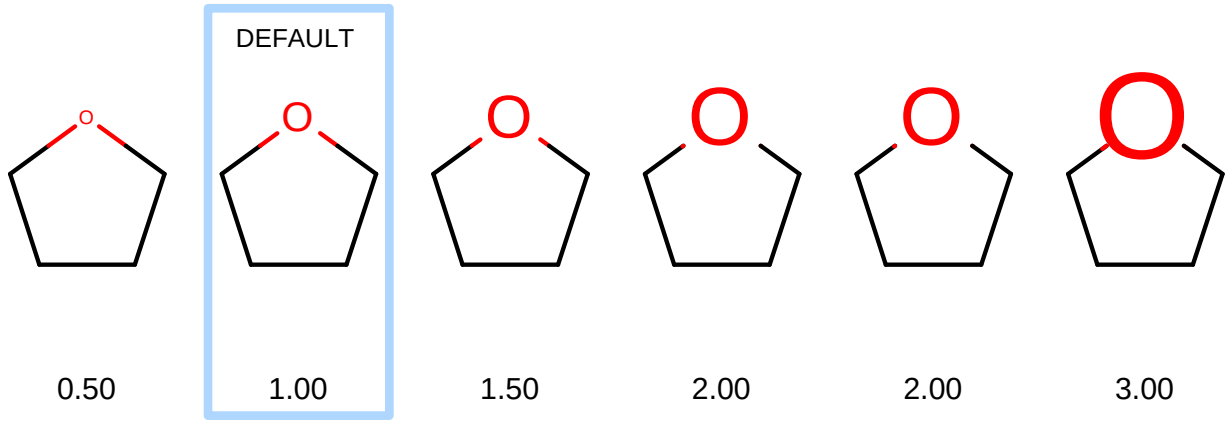


Fig. 19: **Examples of scaling the font of the atom label relative to the molecule**

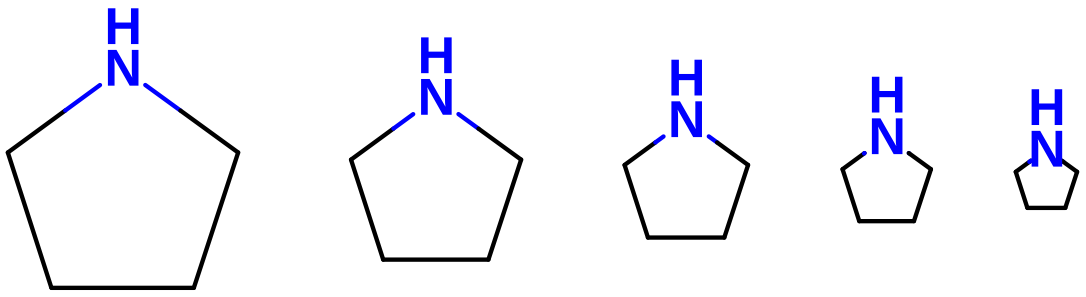


Fig. 20: **Example of atom labels with fixed font size**

Example: (*Figure: Example of using the SetAtomLabelFontScale method*)

```
double width = 300.0;
double height = 200.0;
```

(continues on next page)

(continued from previous page)

```
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetAtomLabelFontScale(1.5);
```

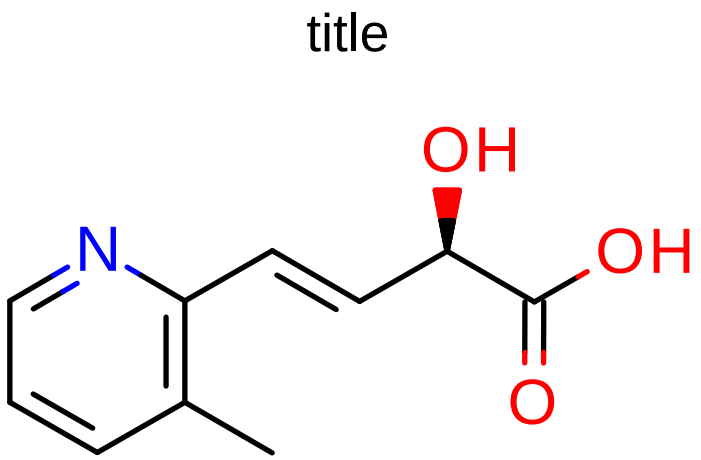


Fig. 21: **Example of using the SetAtomLabelFontScale method**

See also

- *OE2DMolDisplayOptions.GetAtomLabelFontScale* method

SetAtomPropLabelFont

```
void SetAtomPropLabelFont(const OEFont& font)
```

Sets the font that is used to initialize the depiction style of the atom property labels (*OE2DAtomDisplay.GetPropertyFont*).

Example: (*Figure: Example of using the SetAtomPropLabelFont method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
OEFont font = new OEFont();
font.SetStyle(OEFontStyle.Bold);
font.SetColor(oechem.getOEDarkGreen());
opts.SetAtomPropLabelFont(font);
opts.SetAtomPropertyFunctor(new OEDisplayAtomIdx());
```

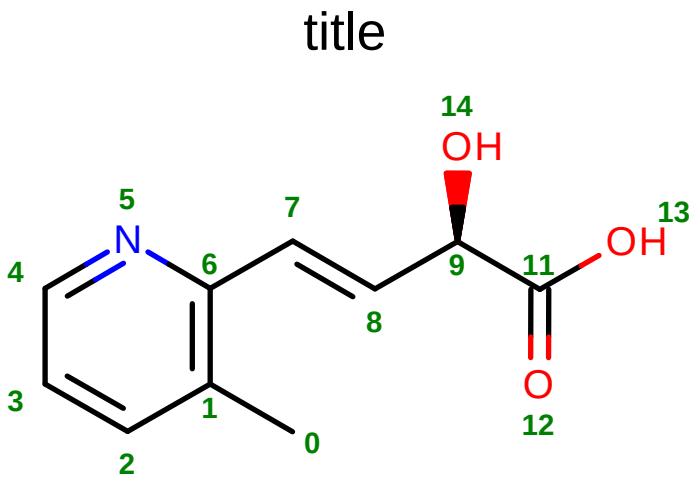


Fig. 22: **Example of using the SetAtomPropLabelFont method**

Note

- The size of the atom property label fonts also depends on the scaling factor used to fit a molecule to a given dimension (*OE2DMolDisplayOptions.GetScale*) and the multiplier set by the *OE2DMolDisplayOptions.SetAtomPropLabelFontScale* method. See example in *Figure: Example of scaling the font of the atom property labels along with the molecule*

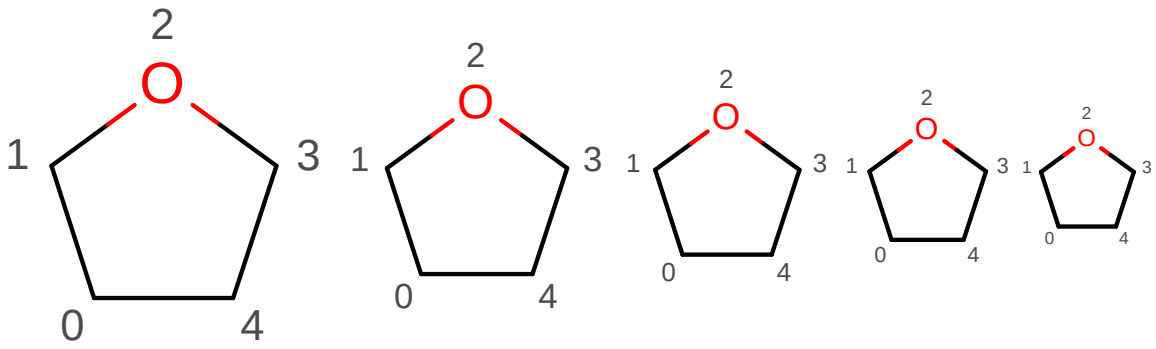


Fig. 23: **Example of scaling the font of the atom property labels along with the molecule**

See also

- *OE2DMolDisplayOptions.GetAtomPropLabelFont* method
- *OEFont* class

SetAtomPropLabelFontScale

```
void SetAtomPropLabelFontScale(double scale)
```

Sets the multiplier that can be used increase or decrease the size of the fonts of the atom property labels.

scale

This value has to be in the range of `[0.5, 2.0]`. See examples in *Figure: Examples of scaling the font of the atom property labels relative to the molecule*

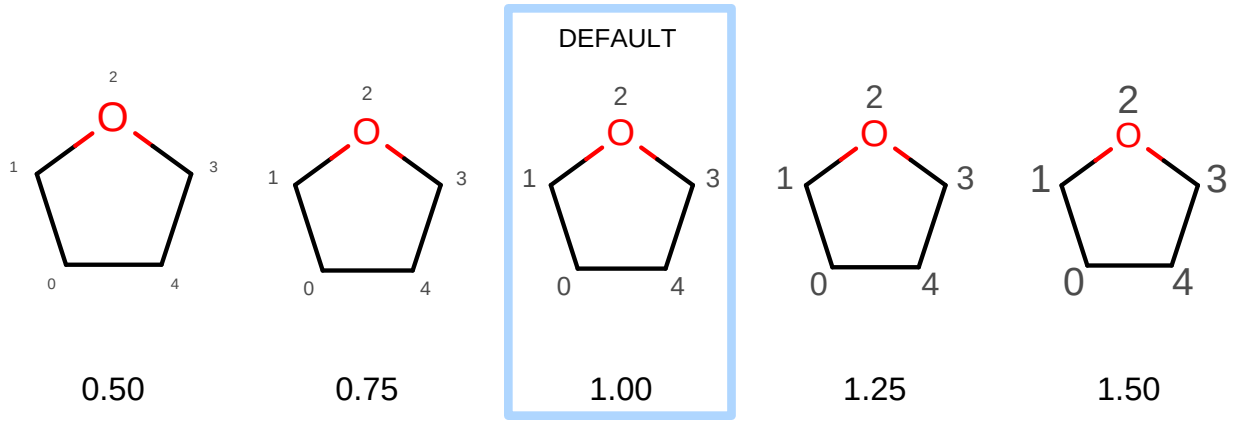


Fig. 24: **Examples of scaling the font of the atom property labels relative to the molecule**

Example: (*Figure: Example of using the SetAtomPropLabelFontScale method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetAtomPropLabelFontScale(1.5);
opts.SetAtomPropertyFunctor(new OEDisplayAtomIdx());
```

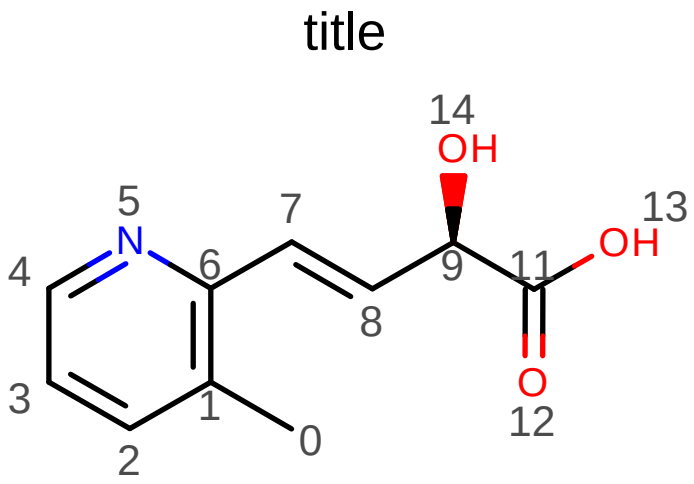


Fig. 25: **Example of using the SetAtomPropLabelFontScale method**

See also

- *OE2DMolDisplayOptions.GetAtomPropLabelFontScale* method

SetAtomPropertyFunctor

```
void SetAtomPropertyFunctor(const OEDisplayAtomPropBase& func)
```

Sets the functor that is used to initialize the atom property labels (*OE2DAtomDisplay.GetProperty*)

Example: (*Figure: Example of using the SetAtomPropertyFunctor method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetAtomPropertyFunctor(new OEDisplayAtomIdx());
```

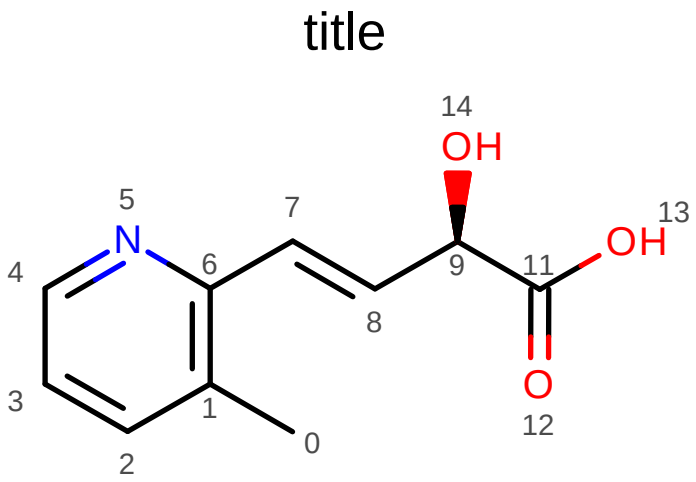


Fig. 26: **Example of using the SetAtomPropertyFunctor method**

Hint

It is recommended to use the *OE2DMolDisplayOptions.SetAtomPropertyFunctor* method to define the atom property labels when the *OE2DMolDisplay* object is constructed. All labels displayed on the molecule diagram have to be known in advance in order to be able to minimize the number of label clashes and clippings when calculating the positions of the bond property labels.

See example of user-defined atom properties in *Displaying Atom Properties* section.

See also

- *OE2DMolDisplayOptions.GetAtomPropertyFunctor* method
- *OEDisplayAtomPropBase* class
- *OEDisplayAtomIdx* class
- *OEDisplayAtomMapIdx* class
- *OEDisplayNoAtomProp* class

SetAtomStereoStyle

```
void SetAtomStereoStyle(unsigned int style)
```

Sets the style that controls what atom stereo information is displayed.

style
This value has to be from the *OEAtomStereoStyle* namespace.

Example: (*Figure: Example of using the SetAtomStereoStyle method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetAtomStereoStyle(OEAtomStereoStyle.Display.All|OEAtomStereoStyle.HashWedgeStyle.
↪Standard);
```

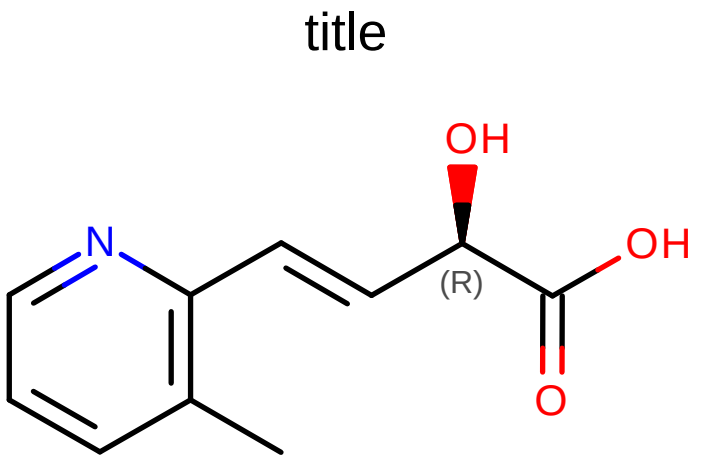


Fig. 27: **Example of using the SetAtomStereoStyle method**

See also

- *OE2DMolDisplayOptions.GetAtomStereoStyle* method
- *OEAtomStereoStyle* namespace

SetAtomSVGMarkupFunctor

```
void SetAtomSVGMarkupFunctor(const OEAtomSVGMarkupBase& func)
```

Sets the functor that defines how atoms are marked in `svg` image. Drawing elements representing atoms in `svg` image are grouped together in the following format in which the `<group id>` and `<class name>` strings are defined by the given functor:

```
<g id='<group id>' class='<class name>'>
..
list of drawing elements
..
</g>
```

Note

This setting has only effect when generating `.svg` images.

See also

- *OEAtomSVGMarkupBase* abstract base class
- *OEAtomSVGNoMarkup* class
- *OEAtomSVGAtomIdxMarkup* class
- *OEAtomSVGResidueMarkup* class
- *OE2DMolDisplayOptions.GetAtomSVGMarkupFunctor* method

SetAtomVisibilityFunctor

```
void SetAtomVisibilityFunctor(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& func)
```

Sets the functor that is used to identify visible atoms.

Example: (*Figure: Example of using the SetAtomVisibilityFunctor method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetAtomVisibilityFunctor(new OEAtomIsInRing());
```

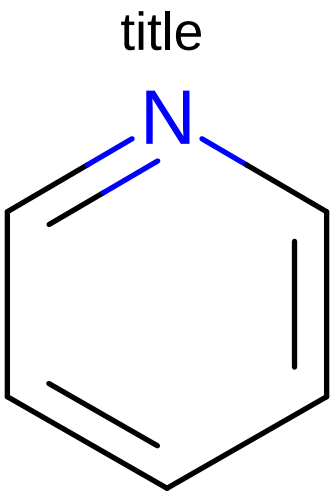


Fig. 28: **Example of using the SetAtomVisibilityFunctor method to only display rings**

Hint

It is recommended to use the *OE2DMolDisplayOptions.SetAtomVisibilityFunctor* method to identify the visible atoms when the *OE2DMolDisplay* object is constructed. All objects displayed in the molecule diagram have to be known in advance in order to be able to correctly size and scale the molecule.

See additional examples of atom visibility control in the *Displaying Atom Properties* section.

See also

- *OE2DMolDisplayOptions.GetAtomVisibilityFunctor* method
- OEUnaryAtomPred

SetBackgroundColor

```
void SetBackgroundColor(const OESystem::OEColor& color)
```

Sets the color that is used to clear the background of an image (by calling *OEImageBase.Clear* method) before rendering the molecule.

Example: (*Figure: Example of using the SetBackgroundColor method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetBackgroundColor(oechem.getOEYellowTint());
```

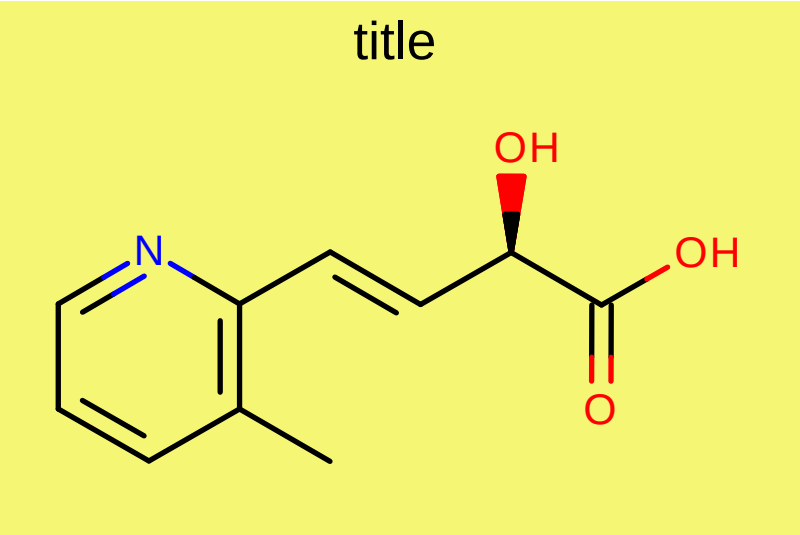


Fig. 29: **Example of using the SetBackgroundColor method**

See also

- *OE2DMolDisplayOptions.GetBackgroundColor* method
- OEColor class

SetBondColorStyle

```
void SetBondColorStyle(unsigned int style)
```

Sets the style that controls how bonds are colored.

style
This value has to be from the *OEBondColorStyle* namespace.

Example: (*Figure: Example of using the SetBondColorStyle method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetBondColorStyle(OEBondColorStyle.Monochrome);
```

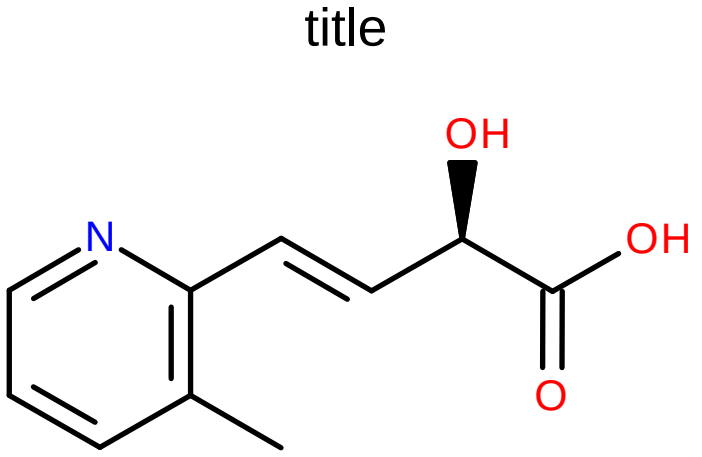


Fig. 30: **Example of using the SetBondColorStyle method**

See also

- *OE2DMolDisplayOptions.GetBondColorStyle* method
- *OEBondColorStyle* namespace

SetBondLineGapScale

```
void SetBondLineGapScale(double scale)
```

Sets the multiplier that can be used increase or decrease the gap between the lines of double and triple bonds.

scale

This value has to be in the range of `[0.5, 2.0]`. See examples in *Figure: Examples of altering the gap between the lines of double and triple bonds*

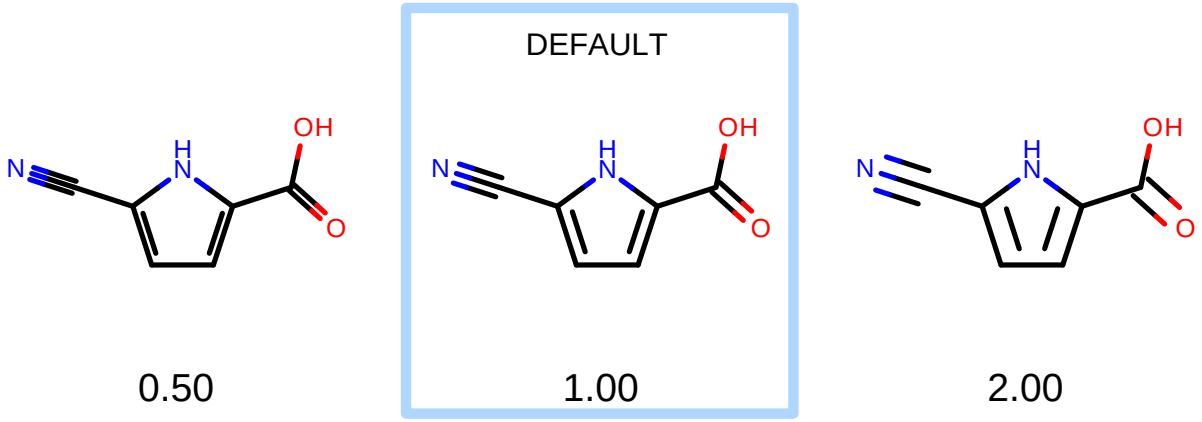


Fig. 31: **Examples of altering the gap between the lines of double and triple bonds**

Example: (*Figure: Example of using the SetBondLineGapScale method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetBondLineGapScale(0.5);
```

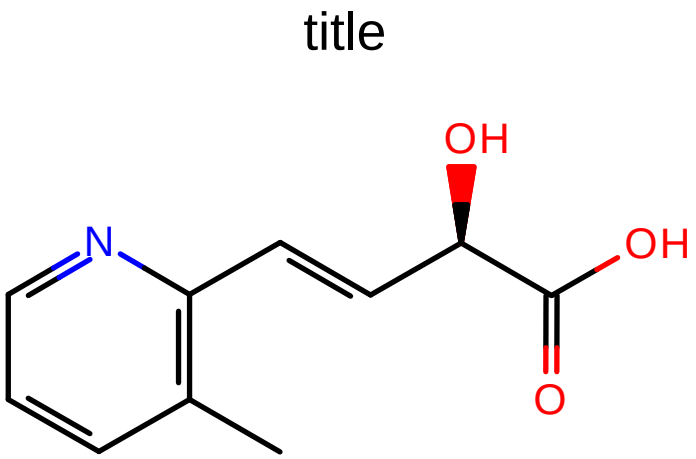


Fig. 32: **Example of using the SetBondLineGapScale method**

See also

- *OE2DMolDisplayOptions.GetBondLineGapScale* method

SetBondLineAtomLabelGapScale

```
void SetBondLineAtomLabelGapScale(double scale)
```

Sets the multiplier that can be used to increase or decrease the gap between the line(s) of bonds and the adjacent atom labels.

scale

This value has to be in the range of `[0.5, 2.0]`. See examples in *Figure: Examples of altering the gap between the line(s) of bonds and adjacent atom labels*

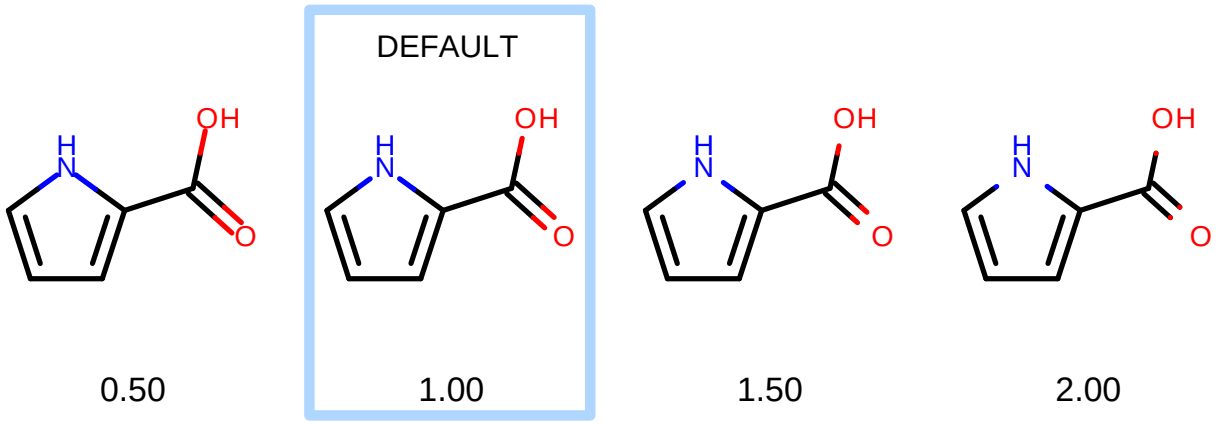


Fig. 33: **Examples of altering the gap between the line(s) of bonds and adjacent atom labels**

Example: (*Figure: Example of using the SetBondLineAtomLabelGapScale method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetBondLineAtomLabelGapScale(2.0);
```

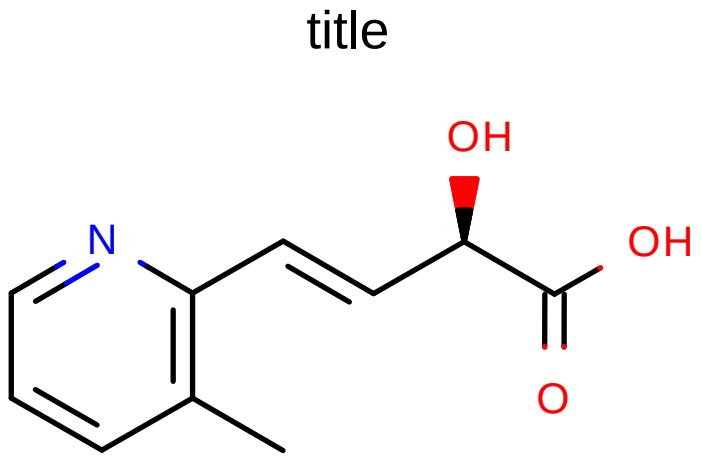


Fig. 34: **Example of using the SetBondLineAtomLabelGapScale method**

See also

- *OE2DMolDisplayOptions.GetBondLineAtomLabelGapScale* method

SetBondPropLabelFont

```
void SetBondPropLabelFont(const OEFont& color)
```

Sets the font that is used to initialize the depiction style of the bond property labels (*OE2DBondDisplay.GetPropertyFont*).

Example: (*Figure: Example of using the SetBondPropLabelFont method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
OEFont font = new OEFont();
font.SetStyle(OEFontStyle.Bold);
font.SetColor(oechem.getOEDarkGreen());
opts.SetBondPropLabelFont(font);
opts.SetBondPropertyFunctor(new OEDisplayBondIdx());
```

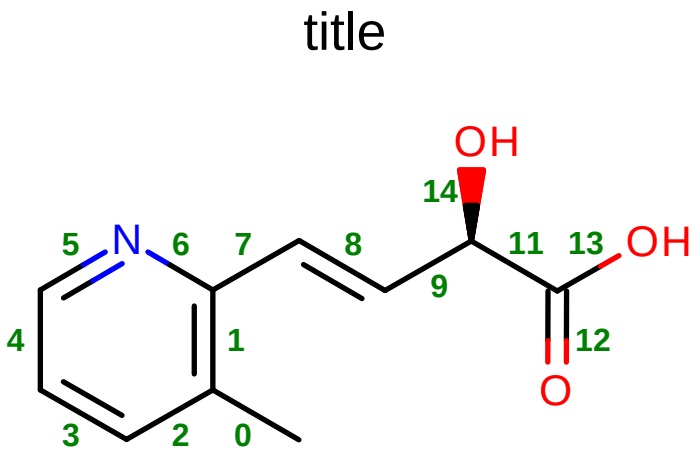


Fig. 35: **Example of using the SetBondPropLabelFont method**

Note

- The size of the bond property label fonts also depends on the scaling factor used to fit a molecule to a given dimension (`OE2DMolDisplayOptions.GetScale`) and the multiplier set by the `OE2DMolDisplayOptions.SetBondPropLabelFontScale` method. See example in *Figure: Example of scaling the font of the bond property labels along with the molecule*

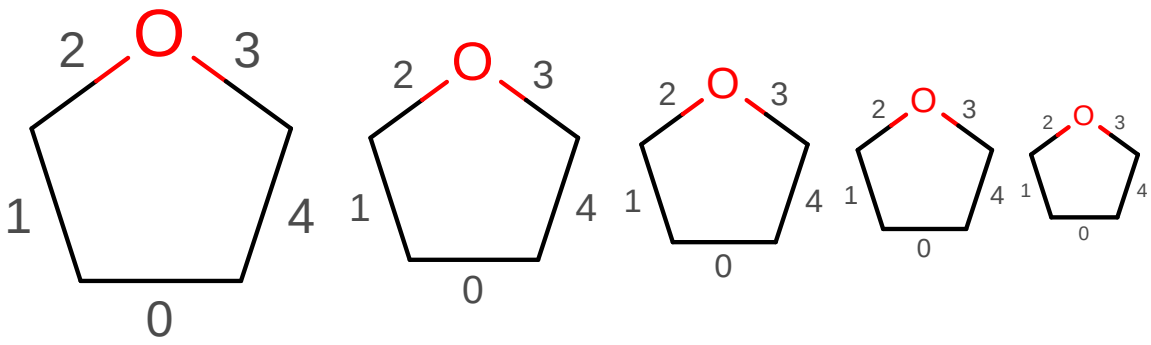


Fig. 36: **Example of scaling the font of the bond property labels along with the molecule**

See also

- *OE2DMolDisplayOptions.GetBondPropLabelFont* method
- *OEFont* class

SetBondPropLabelFontScale

```
void SetBondPropLabelFontScale(double scale)
```

Sets the multiplier that can be used increase or decrease the size of the fonts of the bond property labels.

scale

This value has to be in the range of `[0.5, 2.0]`. See examples in *Figure:*Examples of scaling the font of the bond property labels relative to the molecule SetBondPropLabelFontScale method*

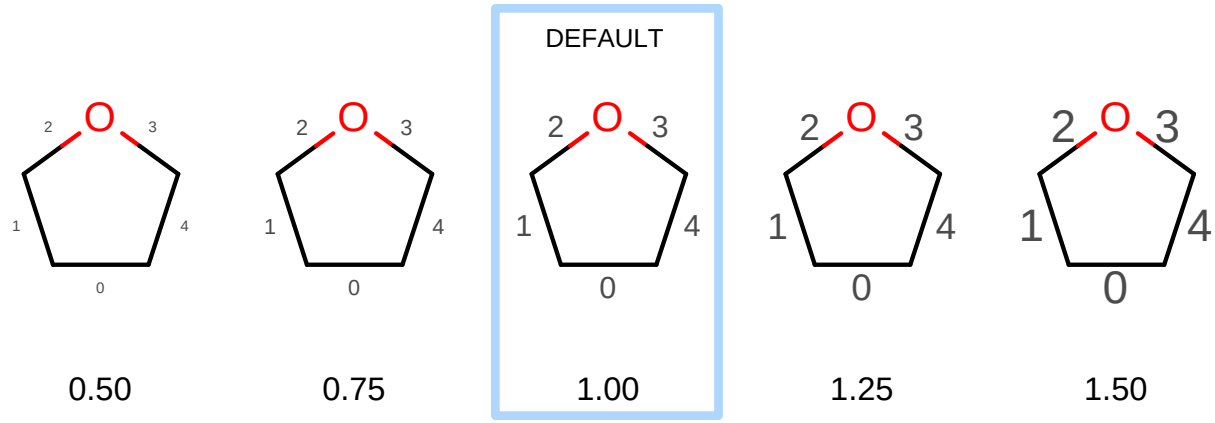


Fig. 37: **Examples of scaling the font of the bond property labels relative to the molecule**

Example: (*Figure: Example of using the SetBondPropLabelFontScale method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetBondPropLabelFontScale(1.5);
opts.SetBondPropertyFunctor(new OEDisplayBondIdx());
```

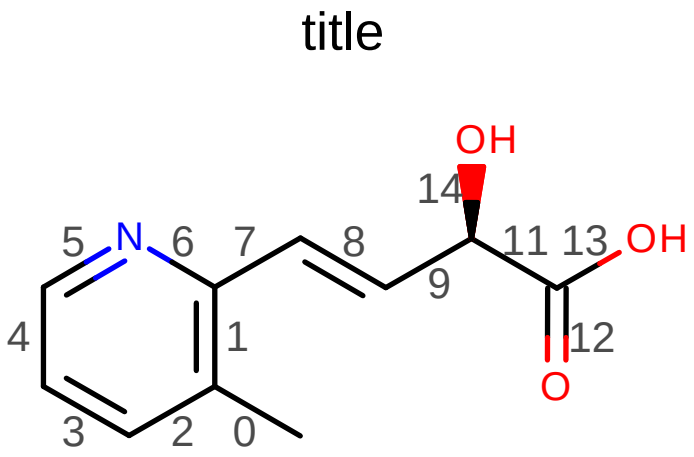


Fig. 38: **Example of using the SetBondPropLabelFontScale method**

See also

- *OE2DMolDisplayOptions.GetBondPropLabelFontScale* method

SetBondPropertyFunctor

```
void SetBondPropertyFunctor(const OEDisplayBondPropBase& func)
```

Sets the functor that is used to initialize the bond property labels (*OE2DBondDisplay.GetProperty*)

Example: (*Figure: Example of using the SetBondPropertyFunctor method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetBondPropertyFunctor(new OEDisplayBondIdx());
```

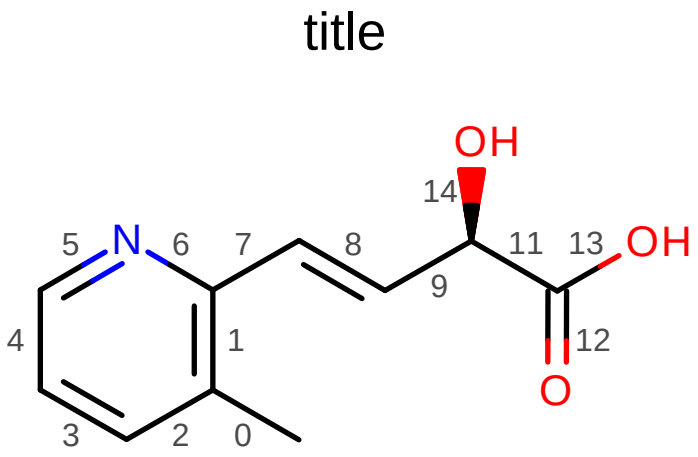


Fig. 39: **Example of using the SetBondPropertyFunctor method**

Hint

It is recommended to use the *OE2DMolDisplayOptions.SetBondPropertyFunctor* method to define the bond property labels when the *OE2DMolDisplay* object is constructed. All labels displayed on the molecule diagram have to be known in advance in order to be able to minimize the number of label clashes and clippings when calculating the positions of the bond property labels.

See example of user-defined bond properties in *Displaying Bond Properties* section.

See also

- *OE2DMolDisplayOptions.GetBondPropertyFunctor* method
- *OEDisplayBondPropBase* class
- *OEDisplayBondIdx* class
- *OEDisplayNoBondProp* class

SetBondStereoStyle

```
void SetBondStereoStyle(unsigned int style)
```

Sets the style that controls what bond stereo information is displayed.

style
This value has to be from the *OEBondStereoStyle* namespace.

Example: (*Figure: Example of using the SetBondStereoStyle method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
```

(continues on next page)

(continued from previous page)

```
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetBondStereoStyle(OEBondStereoStyle.Display.All);
```

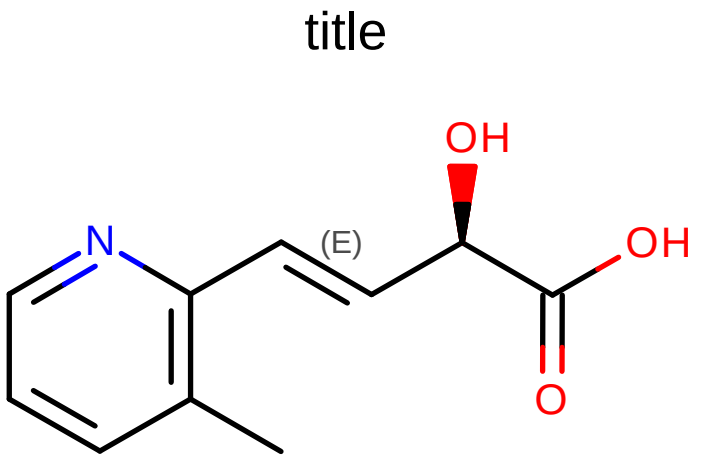


Fig. 40: **Example of using the SetBondStereoStyle method**

See also

- *OE2DMolDisplayOptions.GetBondStereoStyle* method
- *OEBondStereoStyle* namespace

SetBondSVGMarkupFunctor

```
void SetBondSVGMarkupFunctor(const OEBondSVGMarkupBase& func)
```

Sets the functor that defines how bonds are marked in `svg` image. Drawing elements representing bonds in `svg` image are grouped together in the following format in which the `<group id>` and `<class name>` strings are defined by the given functor:

```
<g id='<group id>' class='<class name>'>
..
list of drawing elements
..
</g>
```

Note

This setting has only effect when generating `.svg` images.

See also

- *OEBondSVGMarkupBase* abstract base class
- *OEBondSVGNoMarkup* class
- *OEBondSVGBondIdxMarkup* class
- `OE2DMolDisplayOptions.GetBondSVGMarkupFunctor` method

SetBondWidthScaling

```
void SetBondWidthScaling(bool scale)
```

Sets whether the line width of the bond are increase or decreased based on the molecule scaling factor (`OE2DMolDisplayOptions.GetScale`). See examples in *Figure: Examples of bond line width*.

Note

- The line width of the bonds also depends on the default bond pen that can be set by using the `OE2DMolDisplayOptions.SetDefaultBondPen` method.

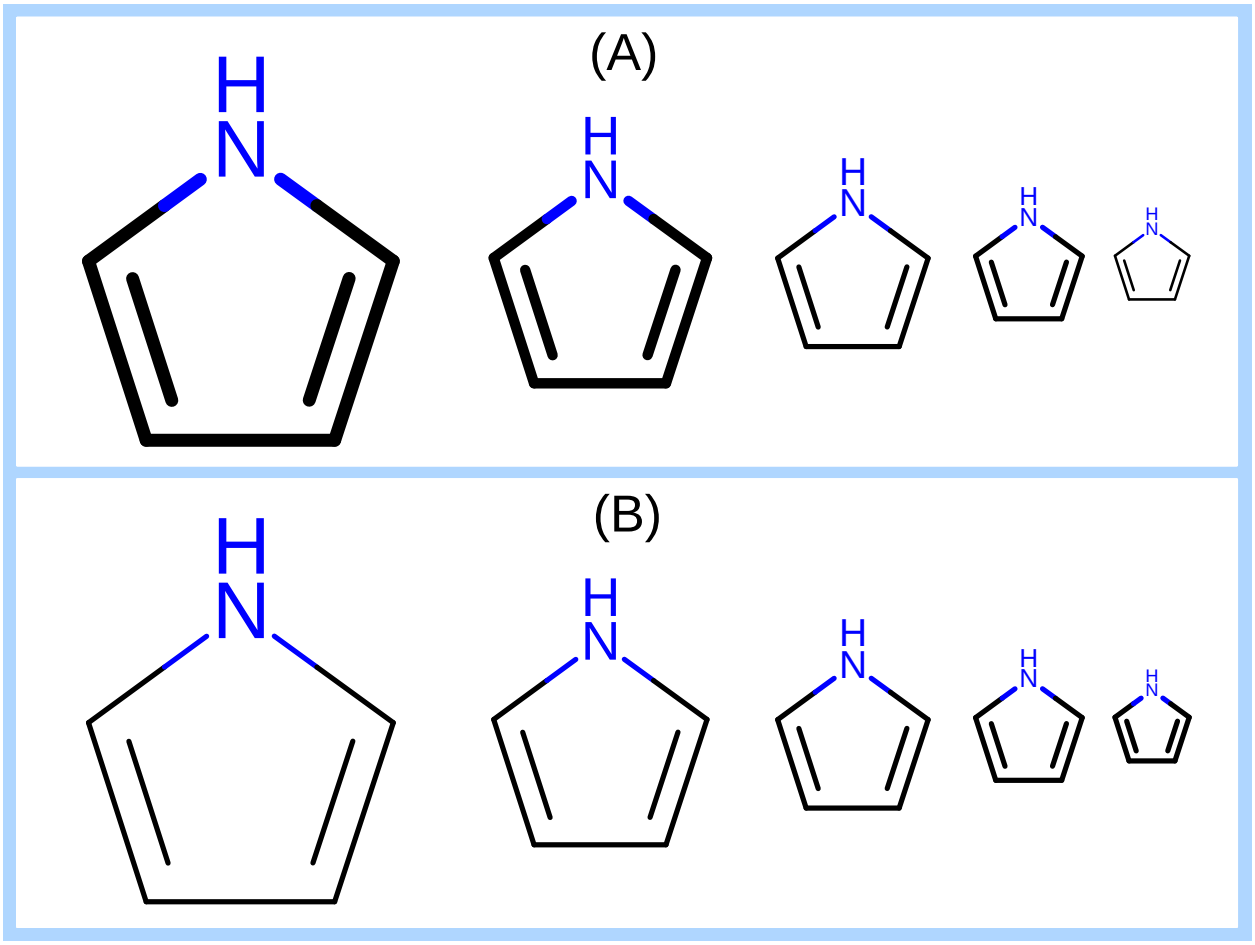


Fig. 41: **Examples of bond line width: (A) scaling the line width of the bonds relative to the molecule; (B) using constant line width that does not depend on the molecule scaling**

Example: (*Figure: Example of using the SetBondWithScaling method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetBondWidthScaling(true);
```

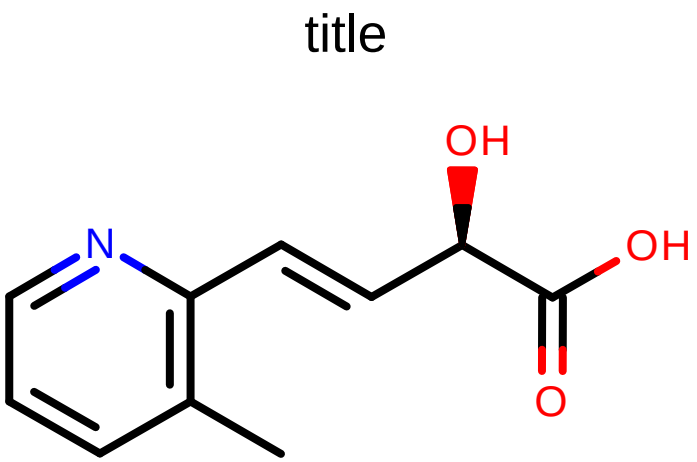


Fig. 42: **Example of using the SetBondWidthScaling method**

See also

- *OE2DMolDisplayOptions.GetBondWidthScaling* method

SetDefaultBondPen

```
void SetDefaultBondPen(const OEPen& pen)
```

Sets the pen that is used to initialize the bond pens (*OE2DBondDisplay.GetBgnPen* and *OE2DBondDisplay.GetEndPen*)

Example: (*Figure: Example of using the SetDefaultBondPen method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
OEPen pen = new OEPen(oechem.getOEBlack(), oechem.getOEBlack(), OEFill.On, 5.0);
opts.SetDefaultBondPen(pen);
```

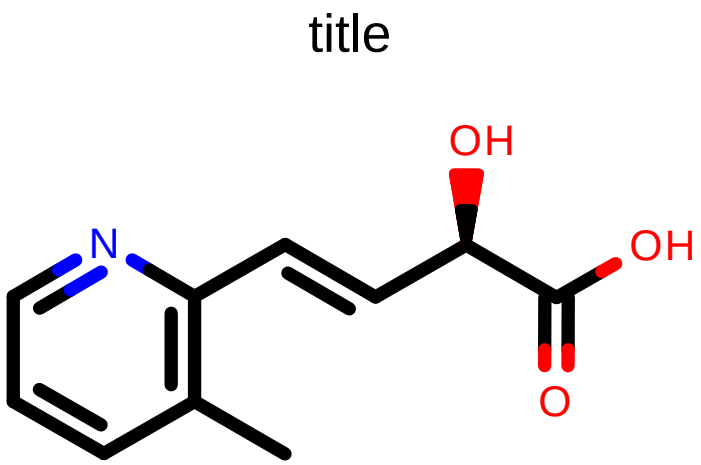


Fig. 43: **Example of using the SetDefaultBondPen method**

Note

- The style of the pens being used to draw bonds also depends on the bond display style (*OE2DBondDisplay*)
- The color of the pens being used to draw bonds also depends on the given atom and bond color styles (`OE2DMolDisplayOptions.SetAtomColorStyle` and `OE2DMolDisplayOptions.SetBondColorStyle`, respectively)

See also

- `OE2DMolDisplayOptions.GetDefaultBondPen` method
- *OEPen* class

SetDimensions

```
void SetDimensions(double width, double height, double scale)
```

Sets parameters that control the size of the depicted molecule.

width
A non-negative number that controls the horizontal limit.

height
A non-negative number that controls the vertical limit.

scale
A non-negative number that controls the magnification of the depicted molecule. `OEScale.AutoScale` scaling means that the molecule is scaled to maximally fit the given dimensions.

Example: (*Figure: Example of using the SetDimensions method*)

```
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions();
double width  = 180.0;
double height = 180.0;
double scale  = 20.0;
opts.SetDimensions(width, height, scale);
```

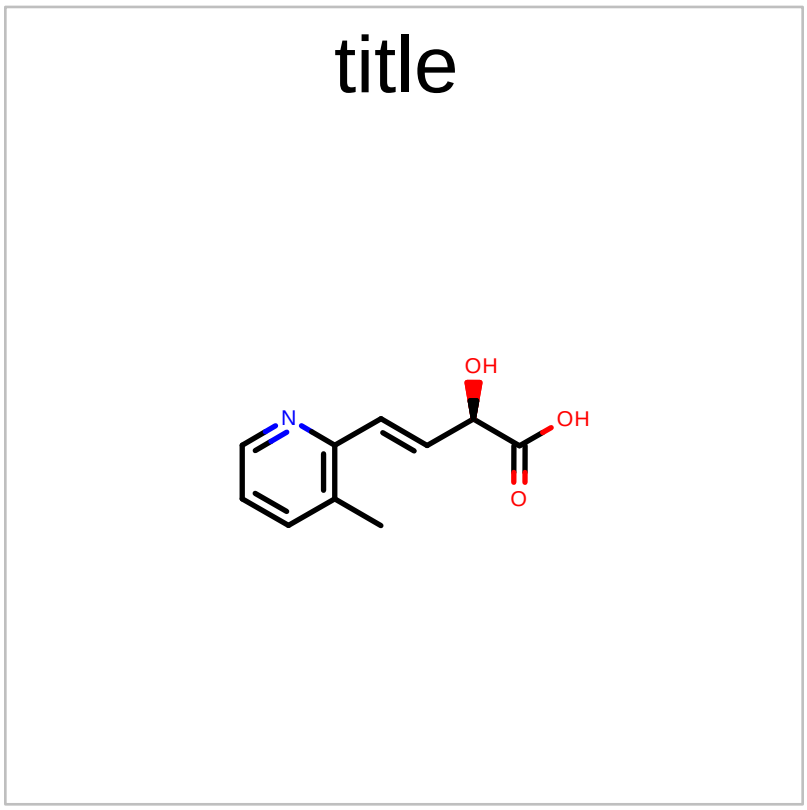


Fig. 44: **Example of using the SetDimensions method**

See also

- *OE2DMolDisplayOptions.GetHeight* and *OE2DMolDisplayOptions.SetHeight* method
- *OE2DMolDisplayOptions.GetScale* and *OE2DMolDisplayOptions.SetScale* method
- *OE2DMolDisplayOptions.GetWidth* and *OE2DMolDisplayOptions.SetWidth* method
- *Controlling the Size of Depicted Molecules* section

SetExplicitAtomLabelAngle

```
void SetExplicitAtomLabelAngle(const double angle)
```

In the case of an almost linear X-C-X bond, an explicit "C" label will be drawn if the angle is more that the given angle. 180.0 degree means that the "C" label will never be drawn.

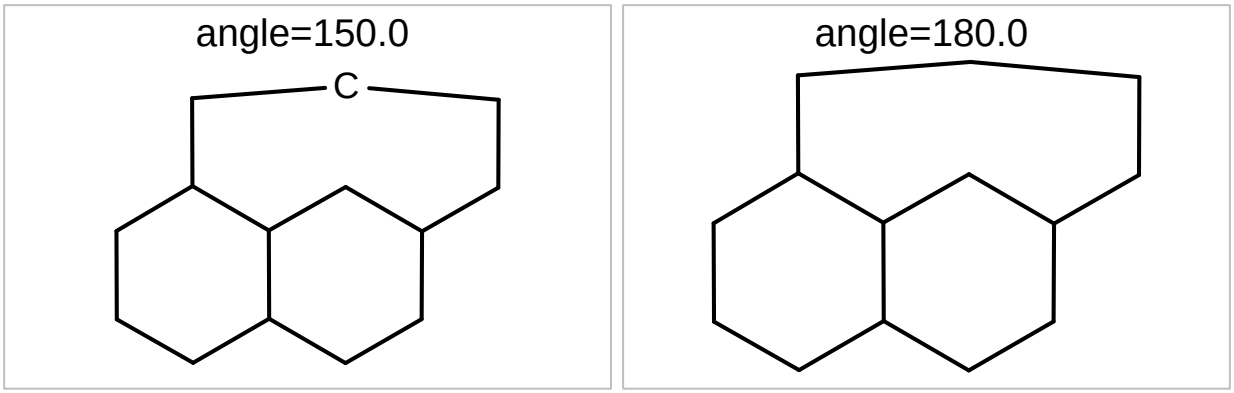


Fig. 45: **Example of using the SetExplicitAtomLabelAngle method**

See also

- *OE2DMolDisplayOptions.GetExplicitAtomLabelAngle* method

SetHeight

```
void SetHeight(double height)
```

Sets the vertical limit into which the molecule has to be fitted.

height
This number has to be non-negative number.

Example: (*Figure: Example of using the SetHeight method*)

```
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions();
opts.SetHeight(150.0);
opts.SetBackgroundColor(oechem.getOELightGrey());
```

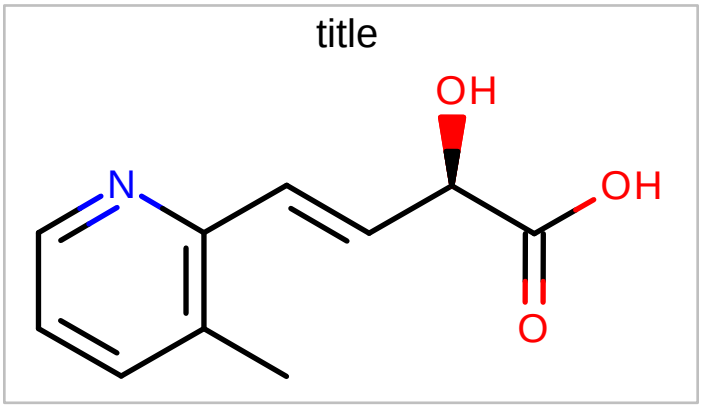


Fig. 46: **Example of using the SetHeight method**

See also

- *OE2DMolDisplayOptions.GetHeight* method
- *OE2DMolDisplayOptions.SetDimensions* method
- *Controlling the Size of Depicted Molecules* section

SetHydrogenStyle

```
void SetHydrogenStyle(unsigned int style)
```

Sets the style that controls how implicit and explicit hydrogens are displayed.

style
This value has to be from the *OEHydrogenStyle* namespace.

Example: (*Figure: Example of using the SetHydrogenStyle method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetHydrogenStyle(OEHydrogenStyle.ImplicitTerminal);
```

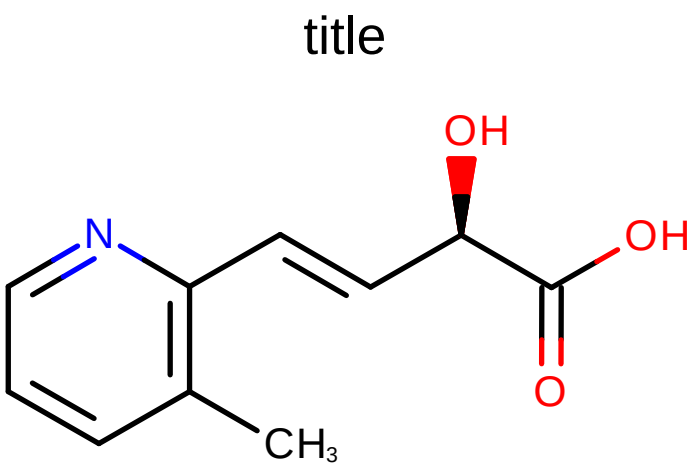


Fig. 47: **Example of using the SetHydrogenStyle method**

See also

- *OE2DMolDisplayOptions.GetHydrogenStyle* method
- *OEHydrogenStyle* namespace

SetMargin

```
bool SetMargin(unsigned int marginloc, double margin)
```

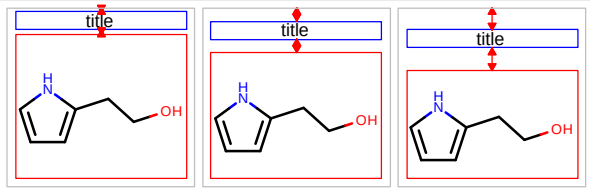
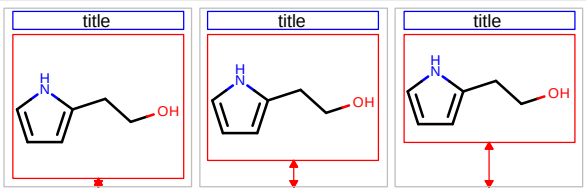
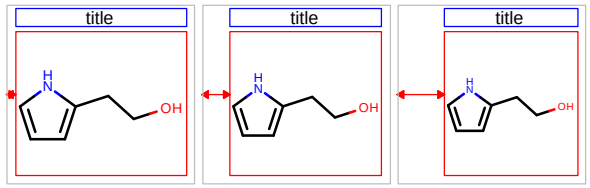
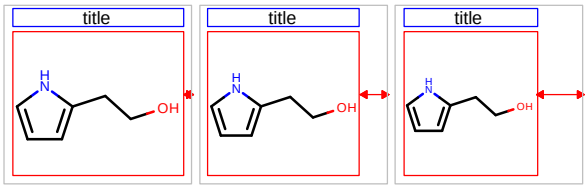
Sets the size of a specific margin of the *OE2DMolDisplayOptions* object.

marginloc
This value has to be from the *OEMargin* namespace.

margin
This number is considered as a percentage of either the width or the height of the molecule display and has to be in the range of `[2.5%, 25.0%]`. See example in *Table: Example of setting specific margins to 5% (default), 15% and 25%*. For example, when setting a margin to 15.0%:

- in case of the **top** or the **bottom** margins, it means that the margin will be 15% of the total height of the molecule display
- in case of the **left** or the **right** margins, it means that the margin will be 15% of the total width of the molecule display

Table 4: **Example of setting specific margins to 5% (default), 15% and 25%**

top margin	bottom margin
	
left margin	**right margin**
	

Example: (*Figure: Example of using the SetMargin method*)

```
double width = 240.0;
double height = 180.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetMargin(OEMargin.Bottom, 25.0);
```

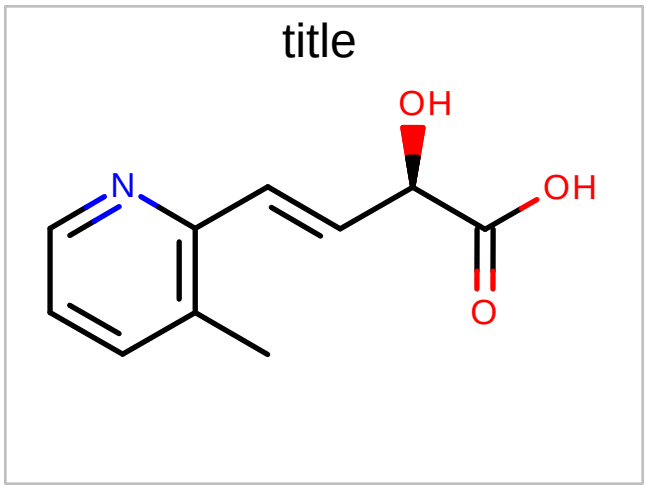


Fig. 48: **Example of using the SetMargin method**

See also

- *OE2DMolDisplayOptions.GetMargin* method
- *OEMargin* namespace

SetMargins

```
bool SetMargins(double margin)
```

Sets the size of all margins of the *OE2DMolDisplayOptions* object.

margin

This number is considered as a percentage of either the width or the height of the molecule display and has to be in the range of `[2.5%, 25.0%]`. For example, 15.0% means, that the left and right margin are 15% of the total width of the molecule display, and the top and bottom margins are 15% of the total height of the molecule display. See example in *Figure: Example of setting all margins to 5% (default), 15% and 25%*.

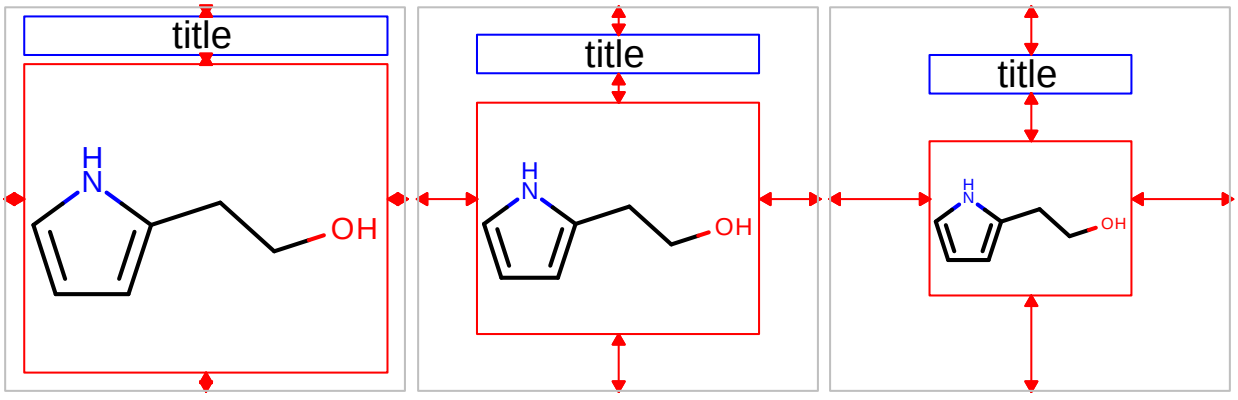


Fig. 49: **Example of setting all margins to 5% (default), 15% and 25%**

Example: (*Figure: Example of using the SetMargin method*)

```
double width = 240.0;
double height = 180.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetMargins(25.0);
```

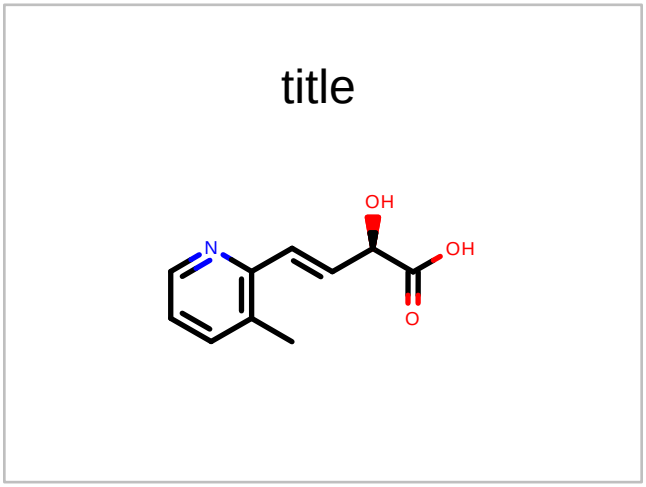


Fig. 50: **Example of using the SetMargins method**

See also

- *OE2DMolDisplayOptions.GetMargin* method
- *OEMargin* namespace

SetProtectiveGroupLabelFont

```
void SetProtectiveGroupLabelFont(const OEFont& font)
```

Sets the font that is used to display the abbreviations of contracted functional groups.

Example: (*Figure: Example of using the ProtectiveGroupLabelFont method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
OEFont font = new OEFont();
font.SetColor(oechem.getOEDarkRed());
opts.SetProtectiveGroupLabelFont(font);
opts.SetProtectiveGroupStyle(OEProtectiveGroupStyle.All);
```

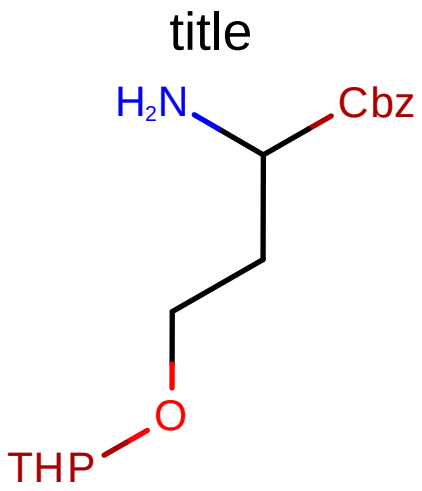


Fig. 51: **Example of using the SetProtectiveGroupLabelFont method**

See also

- *OE2DMolDisplayOptions.GetProtectiveGroupLabelFont* method
- *OE2DMolDisplayOptions.GetProtectiveGroupStyle* method
- *OE2DMolDisplayOptions.GetSuperAtomStyle* method
- *OEFont* class

Note

- The size of the protective group fonts also depends on the scaling factor used to fit a molecule to a given dimension (*OE2DMolDisplayOptions.SetScale*).

SetProtectiveGroupStyle

```
void SetProtectiveGroupStyle(unsigned int style)
```

Sets the style that controls whether or not specific pre-defined protective group are contracted and labeled with corresponding abbreviations.

style
This value has to be from the *OEProtectiveGroupStyle* namespace.

Example: (*Figure: Example of using the SetProtectiveGroupStyle method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetProtectiveGroupStyle(OEProtectiveGroupStyle.All);
```

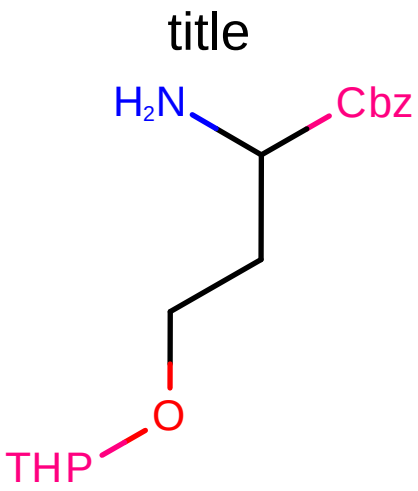


Fig. 52: **Example of using the SetProtectiveGroupStyle method**

See also

- *OE2DMolDisplayOptions.HasProtectiveGroupStyle* method
- *OE2DMolDisplayOptions.GetProtectiveGroupLabelFont* method
- *OE2DMolDisplayOptions.GetSuperAtomStyle* method
- *OEProtectiveGroupStyle* namespace

SetScale

```
void SetScale(double scale)
```

Sets the scaling factor used to magnify a molecule.

scale
This number has to be non-negative number.

Example: (*Figure: Example of using the SetScale method*)

```
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions();
opts.SetScale(20.0);
opts.SetBackgroundColor(oechem.getOELightGrey());
```

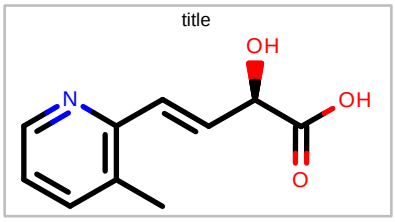


Fig. 53: **Example of using the SetScale method**

See also

- *OE2DMolDisplayOptions.GetScale* method
- *OE2DMolDisplayOptions.SetDimensions* method
- *Controlling the Size of Depicted Molecules* section

SetSuperAtomLabelFont

```
void SetSuperAtomLabelFont(const OEFont& font)
```

Sets the font that is used to display the abbreviations of contracted functional groups.

Example: (*Figure: Example of using the SetSuperAtomLabelFont method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
OEFont font = new OEFont();
font.SetColor(oechem.getOEDarkRed());
opts.SetSuperAtomLabelFont(font);
opts.SetSuperAtomStyle(OESuperAtomStyle.All);
```

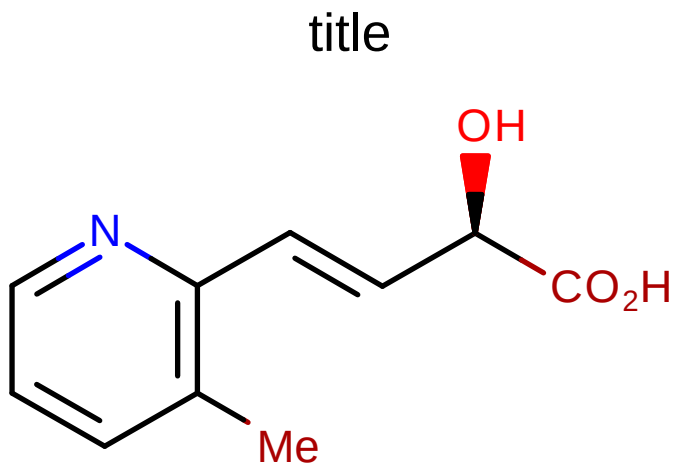


Fig. 54: **Example of using the SetSuperAtomLabelFont method**

See also

- *OE2DMolDisplayOptions.GetSuperAtomLabelFont* method
- *OE2DMolDisplayOptions.GetSuperAtomStyle* method
- *OEFont* class

Note

- The size of the super atom fonts also depends on the scaling factor used to fit a molecule to a given dimension (*OE2DMolDisplayOptions.SetScale*).

SetSuperAtomStyle

```
void SetSuperAtomStyle(unsigned int style)
```

Sets the style that controls whether or not specific pre-defined functional groups are contracted and labeled with corresponding abbreviations.

style
This value has to be from the *OESuperAtomStyle* namespace.

Example: (*Figure: Example of using the SetSuperAtomStyle method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetSuperAtomStyle(OESuperAtomStyle.All);
```

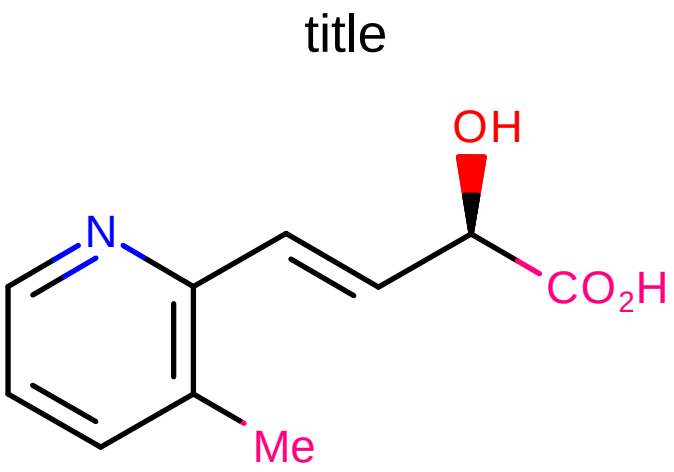


Fig. 55: **Example of using the SetSuperAtomStyle method**

See also

- *OE2DMolDisplayOptions.GetSuperAtomStyle* method
- *OE2DMolDisplayOptions.GetSuperAtomLabelFont* method
- *OESuperAtomStyle* namespace

SetTitleFont

```
void SetTitleFont(const OEFont& font)
```

Sets the font that is used to display the title of the molecule (*i.e.* the string returned by the `OEMolBase.GetTitle` method).

Example: (*Figure: Example of using the SetTitleFont method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
OEFont font = new OEFont();
font.SetStyle(OEFontStyle.Bold);
font.SetColor(oechem.getOEDarkBrown());
opts.SetTitleFont(font);
```

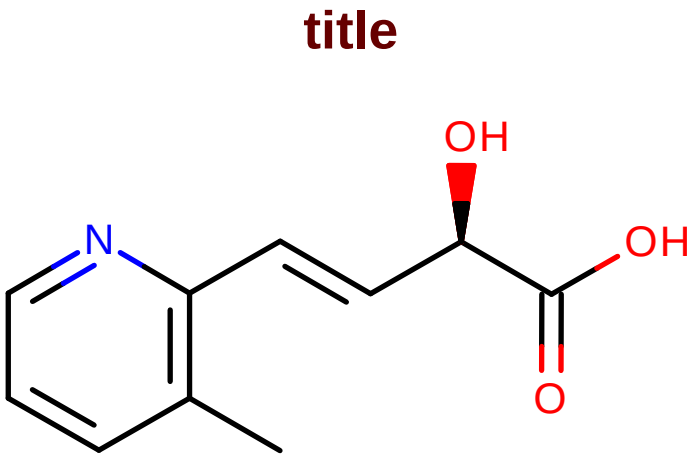


Fig. 56: **Example of using the SetTitleFont method**

Note

The size of the title font also depends on:

- the width and height of the molecule display
- the height of the title area that can be set by *OE2DMolDisplayOptions.SetTitleHeight* method
- the multiplier that can be set by the *OE2DMolDisplayOptions.SetTitleFontScale* method

See example in *Figure: Example of scaling the title along with the molecule*

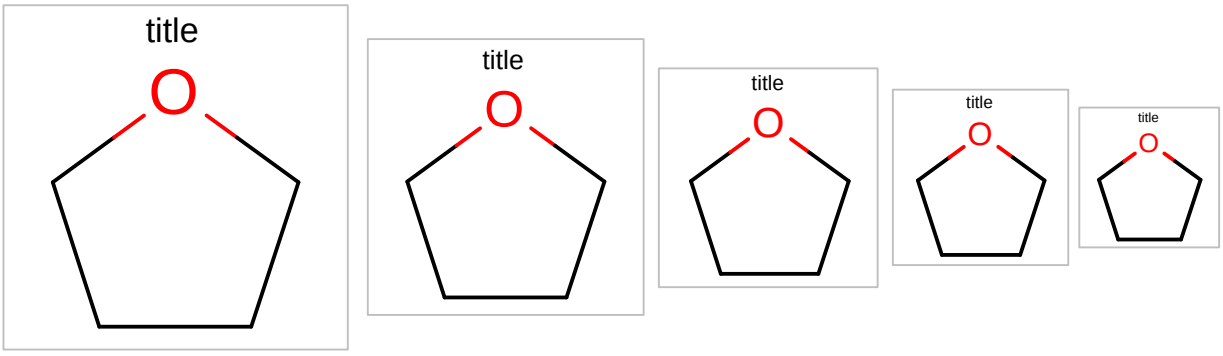


Fig. 57: **Example of scaling the title along with molecule**

See also

- *OE2DMolDisplayOptions.GetTitleFont* method
- *OE2DMolDisplayOptions.SetTitleFontScale* method
- *OE2DMolDisplayOptions.SetTitleHeight* method
- *OEFont* class

SetTitleFontScale

```
void SetTitleFontScale(double scale)
```

Sets the multiplier that can be used increase or decrease the size of the fonts of the atom labels.

scale

This value has to be either `0.0` or in a range of `[0.5, 2.0]`. See examples in *Examples of scaling the font of title relative to the molecule* In case of `0.0`, the title is not scaled with the molecule but rather fixed font size is used regardless of dimensions of the image. See example in *Figure: Example of titles with fixed font size*

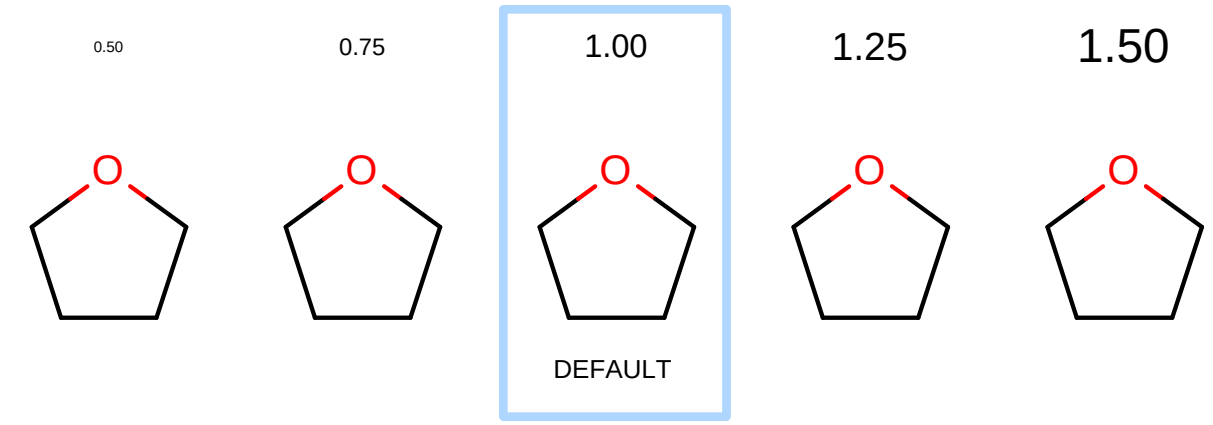


Fig. 58: **Examples of scaling the font of title relative to the molecule**

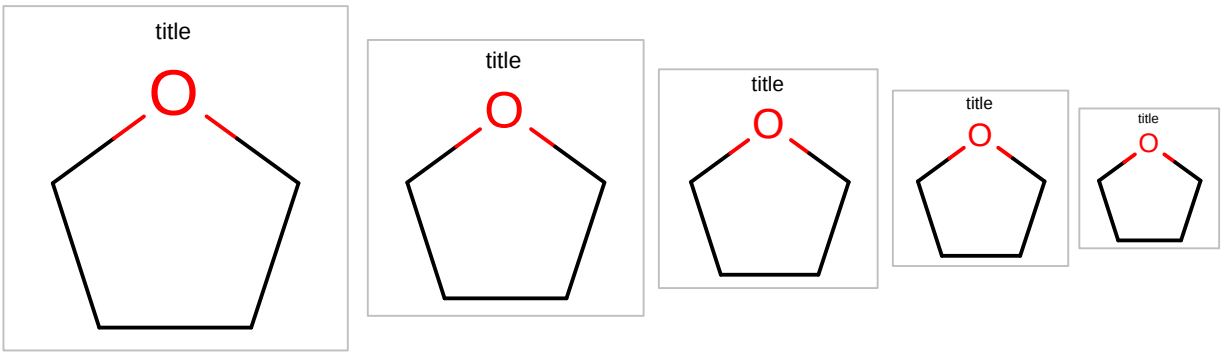


Fig. 59: **Example of titles with fixed font size**

Example: (*Figure: Example of using the SetTitleFontScale method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetTitleFontScale(0.5);
```

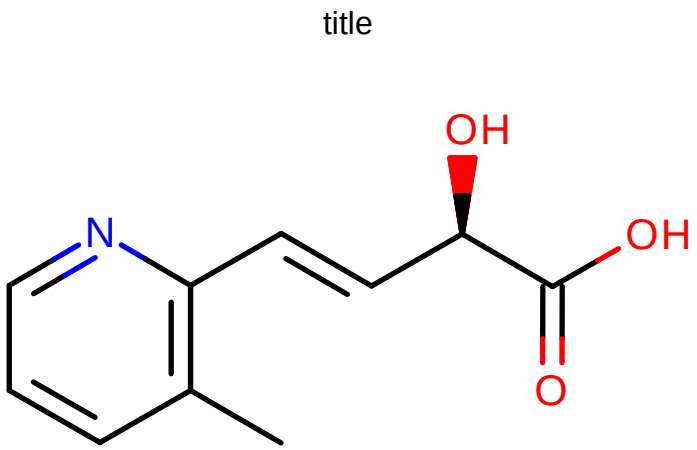


Fig. 60: **Example of using the SetTitleFontScale method**

See also

- *OE2DMolDisplayOptions.GetTitleFontScale* method

SetTitleLocation

```
void SetTitleLocation(unsigned int loc)
```

Sets the position of the molecule title.

loc
This value has to be from the *OETitleLocation* namespace.

Example: (*Figure: Example of using the SetTitleLocation method*)

```
double width = 300.0;
double height = 200.0;
double scale = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetTitleLocation(OETitleLocation.Bottom);
```

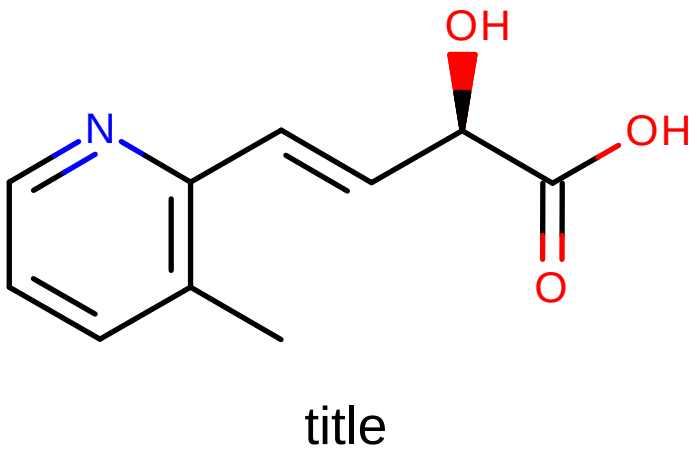


Fig. 61: **Example of using the SetTitleLocation method**

See also

- *OE2DMolDisplayOptions.GetTitleLocation* method
- *OETitleLocation* namespace

SetTitleHeight

```
bool SetTitleHeight(double height)
```

Sets the height of the title area of the *OE2DMolDisplayOptions* object.

height

This number is considered as a percentage of the height of the molecule display and has to be in the range of `[5.0%, 25.0%]`. See example in *Figure: Example of setting height of the title area to 5%, 10% (default) and 25%*.

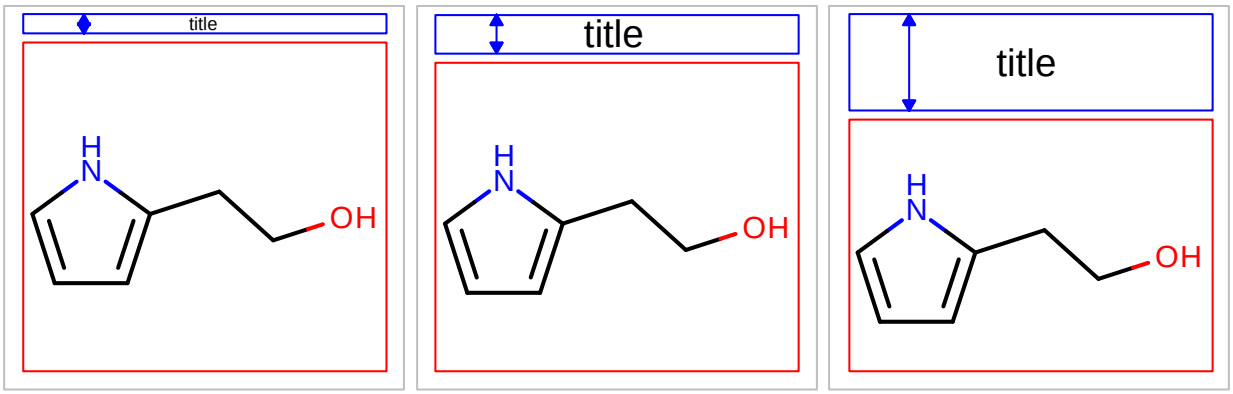


Fig. 62: **Example of setting height of the title area to 5%, 10% (default) and 25%**

Note

A molecule display has a title area only if the title location is not hidden. See also *OETitleLocation* namespace.

Example: (*Figure: Example of using the SetTitleHeight method*)

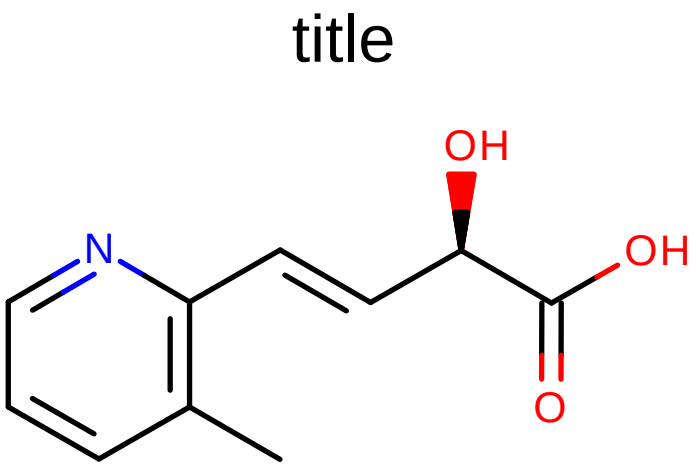


Fig. 63: **Example of using the SetTitleHeight method**

See also

- *OE2DMolDisplayOptions.GetTitleHeight* method

SetWidth

```
void SetWidth(double width)
```

Sets the horizontal limit into which the molecule has to be fitted.

width
: This number has to be non-negative number.

Example: (*Figure: Example of using the SetWidth method*)

```
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions();
opts.SetWidth(150.0);
opts.SetBackgroundColor(oechem.getOELightGrey());
```

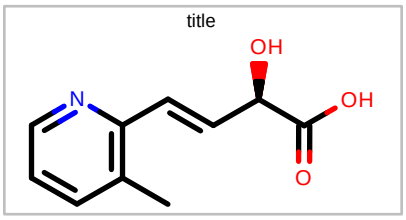


Fig. 64: **Example of using the SetWidth method**

See also

- *OE2DMolDisplayOptions.GetWidth* method
- *OE2DMolDisplayOptions.SetDimensions* method
- *Controlling the Size of Depicted Molecules* section

4.1.5 OE2DPath

```
class OE2DPath
```

A sequence of lines and curves.

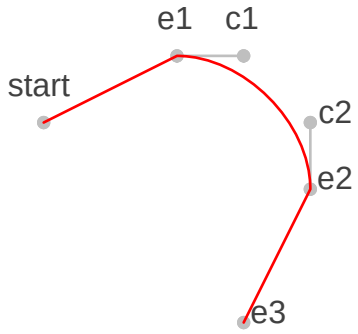


Fig. 65: **Example of path**

See also

- *OEImageBase.DrawPath*

Constructors

```
OE2DPath()
```

Default constructor. The starting point of the path have to be defined using the *OE2DPath.AddStartPoint* method.

```
OE2DPath(const OE2DPoint& start, bool closed = true)
```

Create a new path with a starting point.

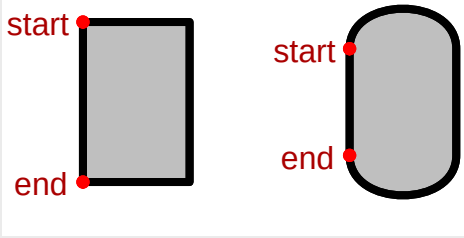
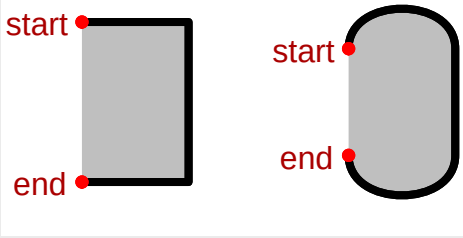
start
The **current point** that will be starting point of the first line or curve.

closed
If true, than a line segment will be added from the starting point to the end point of the last line or curve segment of the path. See examples in *Table: Examples of closed and open paths*.

Note

The starting point of the path will be associated with the *OE2DPathPointType.Start* type when returned by the *OE2DPath.GetPoints* method.

Table 5: **Examples of closed and open paths**

closed paths	open paths
	


```
OE2DPath(const OE2DPath& rhs)
```

Copy constructor.

```
OE2DPath(const OE2DPath& rhs, double scale)
```

Copy constructor with scaling.

scale
The scaling factor is used to create a new image from the given one. The scaling factor has to be a positive (non-zero) number.

```
OE2DPath(const OE2DPath& rhs, const OE2DPoint& offset)
```

Copy constructor with offset.

offset
The offset of new path relative to the copied one.

```
OE2DPath(const OE2DPath& rhs, const OE2DPoint& center,
         double degrees)
```

Copy constructor with rotation.

center
The center of the rotation.

degree
The rotation in degrees.

AddCurveSegment

```
void AddCurveSegment(const OE2DPoint& c1, const OE2DPoint& c2,
                     const OE2DPoint& end)
```

Adds a cubic Bézier curve to the path from the end point of the last line or curve segment of the path to the position **end** using **c1** and **c2** control points.

c1, c2
The control points of the curve.

end
The end point of the curve segment.

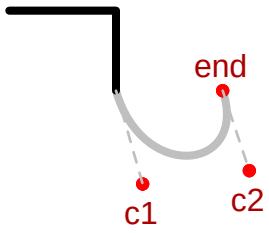


Fig. 66: **Example of adding a curve segment to a path**

Note

Points added to the path with the *OE2DPath.AddCurveSegment* method will be associated with the following types when returned by the *OE2DPath.GetPoints* method:

- *OE2DPathPointType.CurveC1*
- *OE2DPathPointType.CurveC2*
- *OE2DPathPointType.CurveEnd*

See also

- *OE2DPoint* class
- *OEImageBase.DrawCubicBezier* method
- *OE2DPath.GetPoints* method

AddLineSegment

```
void AddLineSegment(const OE2DPoint& end)
```

Adds a line to the path from the end point of the last line or curve segment of the path to the position **end**.

end
The end point of the line segment.

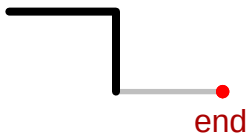


Fig. 67: **Example of adding a line segment to a path**

Note

Points added to the path with the *OE2DPath.AddLineSegment* method will be associated with the *OE2DPathPointType.LineEnd* type when returned by the *OE2DPath.GetPoints* method.

See also

- *OE2DPoint* class
- *OEImageBase.DrawLine* method

AddStartPoint

```
void AddStartPoint(const OE2DPoint& start)
```

Adds the starting point of the path. If clears all previously added line or curve segments.

GetEnd

```
OE2DPoint GetEnd() const
```

Returns the end position of the path.

See also

- *OE2DPath.GetStart* method

GetPoints

```
OESystem::OEIterBase<OE2DPathPoint>* GetPoints() const
```

Returns an iterator over all points of the path. Each point is associated with a type from the *OE2DPathPointType* namespace.

Example:

```
for (OE2DPathPoint p : path.GetPoints()) {
    OE2DPoint pos = new OE2DPoint(p.GetPoint());
    System.out.println(String.format("%3.1f %3.1f %d",
            pos.GetX(), pos.GetY(), p.GetPointType()));
}
```

See also

- *OE2DPathPoint* class
- *OE2DPathPointType* namespace

GetStart

```
OE2DPoint GetStart() const
```

Returns the starting position of the path.

See also

- *OE2DPath.GetEnd* method

IsClosed

```
bool IsClosed() const
```

Returns whether a line segment will be added from the starting point to the end point of the last line or curve segment of the path. See examples in *Table: Examples of closed and open paths*.

See also

- *OE2DPath.SetClosed* method

IsValid

```
bool IsValid() const
```

Returns whether the *OE2DPath* object is valid. A path is considered valid if it has a starting point.

SetClosed

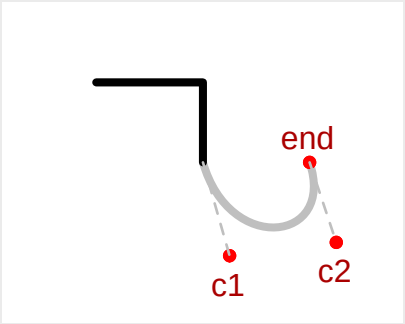
```
void SetClosed(bool closed)
```

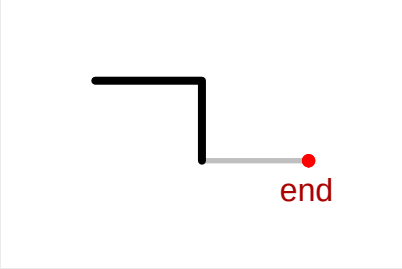
Sets whether a line segment will be added from the starting point to the end point of the last line or curve segment of the path.

4.1.6 OE2DPathPoint

```
class OE2DPathPoint
```

Represent a 2D point of a path with associated type from the *OE2DPathPointType* namespace.





See also

- *OE2DPath.GetPoints* method

Constructors

```
OE2DPathPoint(const OE2DPoint& point, unsigned int ptype)
```

Constructs an *OE2DPathPoint* object with a given type from the *OE2DPathPointType* namespace.

```
OE2DPathPoint(const OE2DPathPoint& rhs)
```

Copy constructor.

GetPoint

```
const OE2DPoint& GetPoint() const
```

Returns the 2D coordinates of the point.

GetPointType

```
unsigned int GetPointType() const
```

Returns the type of the point from the *OE2DPathPointType* namespace.

4.1.7 OE2DPoint

```
class OE2DPoint
```

This class represents *OE2DPoint* that stores an x (horizontal), y (vertical) coordinate pair.

The coordinate system used in **OEDepict TK** has the origin (x=0.0, y=0.0) at the top/left with the x-axis pointing to the right and the y-axis pointing down. See *Figure: 2D coordinate system*.

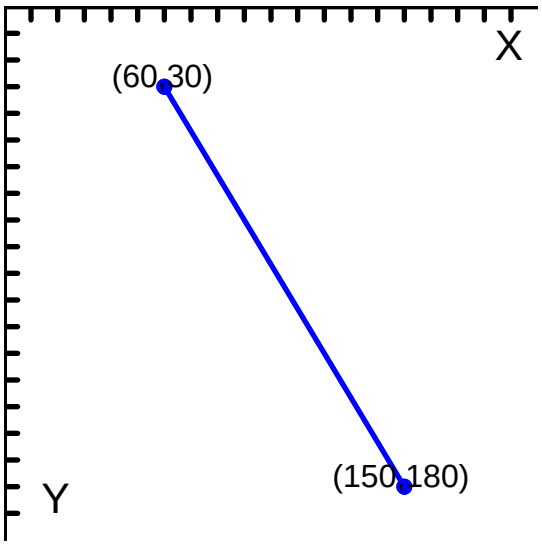


Fig. 68: **2D coordinate system (X = horizontal, Y = vertical)**

Constructors

```
OE2DPoint()
```

Default constructor that initializes an *OE2DPoint* object with x = 0.0, y = 0.0 coordinates.

```
OE2DPoint(double x, double y)
```

Constructor that initializes an *OE2DPoint* object with the give x, y coordinates.

```
OE2DPoint(const OE2DPoint& rhs)
```

Copy constructor.

operator+=

```
OE2DPoint &operator+=(const OE2DPoint &p)
```

Addition assignment operator.

GetX

```
double GetX() const
```

Returns the x (horizontal) coordinate of the *OE2DPoint* object.

See also

- *OE2DPoint.SetY* method

GetY

```
double GetY() const
```

Returns the y (vertical) coordinate of the *OE2DPoint* object.

See also

- *OE2DPoint.SetY* method

SetX

```
void SetX(double x)
```

Sets the x (horizontal) coordinate of the *OE2DPoint* object.

See also

- *OE2DPoint.GetX* method

SetY

```
void SetY(double y)
```

Sets the y (vertical) coordinate of the *OE2DPoint* object.

See also

- *OE2DPoint.GetY* method

4.1.8 OEAlignmentOptions

```
class OEAlignmentOptions
```

This class represents *OEAlignmentOptions* and stores the following parameters which are used when performing molecule alignment.

Property	Get method	Set method
re-generating 2D coordinates	*GetClearCoords*	*SetClearCoords*
adding depiction hydrogens	*GetAddDepictionHydrogens*	*SetAddDepictionHydrogens*
starting from the 2D coords of the reference molecule	*GetFixedCoords*	*SetFixedCoords*
max number of allowed atom pair clushes	*GetMaxAllowedAtomPairCla*	*SetMaxAllowedAtomPairClashes*
max number of bond rotations	*GetMaxBondRotations*	*SetMaxBondRotations*
max number of pattern matches	*GetMaxPatternMatches*	*SetMaxPatternMatches*
RMSD cutoff	*GetRMSDCutoff*	*SetRMSDCutoff*
rotation around bonds	*GetRotateAroundBonds*	*SetRotateAroundBonds*
suppressing hydrogens	*GetSuppressHydrogens*	*SetSuppressHydrogens*

See also

- *OEPrepareAlignedDepiction* function

Constructors

```
OEAlignmentOptions()
```

Default constructor that initializes an *OEAlignmentOptions* object with the following properties:

Property	Default value
re-generating 2D coordinates	true
adding depiction hydrogens	true
starting from the 2D coordinates of the reference molecule	false
max number of allowed atom pair clushes	*0* (means an alignment is rejected if there is any atom clash)
max number of bond rotations	$2^{16} = 65536$ (max allowed number is 2^{31})
max number of pattern matches	*0* (means no limit)
RMSD cutoff	*0.1*
rotation around bonds	true
suppressing hydrogens	true

```
OEAlignmentOptions(bool rotatebonds, bool clearcoords,
                   bool suppressH)
```

Creates an *OEAlignmentOptions* object with the specified parameters.

rotatebonds
: See the *OEAlignmentOptions.SetRotateAroundBonds* method.

clearcoords
: See the *OEAlignmentOptions.SetClearCoords* method.

suppressH
: See the *OEAlignmentOptions.SetSuppressHydrogens* method.

By default, the RMSD cutoff is *0.1* (see the *OEAlignmentOptions.SetRMSDCutoff* method) and maximum number of bond rotations is $2^{16} = 65536$ (see the *OEAlignmentOptions.SetMaxBondRotations* method).

```
OEAlignmentOptions(const OEAlignmentOptions& rhs)
```

Copy constructor.

GetAddDepictionHydrogens

```
bool GetAddDepictionHydrogens() const
```

Returns whether the depiction hydrogens are kept/added during the molecule alignment.

See also

- *OEAlignmentOptions.SetAddDepictionHydrogens* method

GetClearCoords

```
bool GetClearCoords() const
```

Returns whether the 2D coordinates of the *fit* molecule will be re-calculated prior to the alignment process.

See also

- *OEDepictCoordinates* function
- *OEAlignmentOptions.SetClearCoords* method

GetFixedCoords

```
bool GetFixedCoords() const
```

Returns whether the *OEPrepareAlignedDepiction* function can start the alignment process by copying over the coordinates of the *reference* molecule to the *fit* molecule.

See also

- *OEAlignmentOptions.SetFixedCoords* method

GetMaxAllowedAtomPairClashes

```
unsigned int GetMaxAllowedAtomPairClashes() const
```

Returns the maximum number of atom pair clashes allowed in fix coordinate mode.

See also

- *OEAlignmentOptions.SetMaxAllowedAtomPairClashes* method

GetMaxBondRotations

```
unsigned int GetMaxBondRotations() const
```

Returns the maximum number of bond rotations performed during the alignment process.

The default value for maximum number of bond rotations is $2^{16} = 65536$. This means that by default only the first 16 single bonds of the *fit* molecule will be rotated to maximize molecule alignment. The max allowed number is 2^{31} that allows the rotate *31* bonds independently to find the best alignment.

See also

- *OEAlignmentOptions.SetMaxBondRotations* method

GetMaxPatternMatches

```
unsigned int GetMaxPatternMatches() const
```

Returns the maximum number of matches utilized to performed molecule alignments.

See also

- *OEAlignmentOptions.SetMaxPatternMatches* method

GetRMSDCutoff

```
double GetRMSDCutoff() const
```

Returns the RMSD cutoff value used to identify acceptable alignments.

See also

- *OEAlignmentOptions.SetRMSDCutoff* method

GetRotateAroundBonds

```
bool GetRotateAroundBonds() const
```

Returns whether rotation around the bonds of the *fit* molecule is allowed.

See also

- *OEAlignmentOptions.SetRotateAroundBonds* method

GetSuppressHydrogens

```
bool GetSuppressHydrogens() const
```

Returns whether the explicit hydrogens are suppressed in the *fit* molecule prior to the alignment process.

See also

- *OEAlignmentOptions.SetSuppressHydrogens* method

SetAddDepictionHydrogens

```
void SetAddDepictionHydrogens(bool)
```

Sets whether depiction hydrogens are kept/added during alignment.

See also

- *OEAddDepictionHydrogens* function
- *OEPrepareDepictionOptions.GetAddDepictionHydrogens* method

SetClearCoords

```
void SetClearCoords(bool clearcoords)
```

Sets whether the 2D coordinates of the *fit* molecule will be re-calculated prior to the alignment process by calling the *OEPrepareDepiction* function.

See also

- *OEDepictCoordinates* function
- *OEAlignmentOptions.GetClearCoords* method

SetFixedCoords

```
void SetFixedCoords(bool fixed)
```

Sets whether the *OEPrepareAlignedDepiction* function can start the alignment process by copying over the coordinates of the *reference* molecule to the *fit* molecule based on the provided match.

If the alignment process based of 'fixed' coordinates can not be performed or it is unsuccessful *i.e.* no alignment is found without atom clashes than the *OEPrepareAlignedDepiction* function falls back to maximizing the overlap between the *fit* and the *reference* 2D molecules by rotating around single bonds. See also *OEAlignmentOptions.SetMaxAllowedAtomPairClashes* method that can be used to allow some atom clashes.

The alignment process based on 'fixed' coordinates currently can only be performed if:

- neither the *reference* nor the *fit* molecule have more than one component
- there is no partial ring match *i.e.* if a ring atom of the *fit* molecule is matched for alignment than all of the atom of the same ring systems have to be matched for alignment

See also

- *OEAlignmentOptions.GetFixedCoords* method

SetMaxAllowedAtomPairClashes

```
void SetMaxAllowedAtomPairClashes(unsigned int maxclashes)
```

Sets the number of atom pair clashes allowed after the 'fixed' coordinates alignment process (see *OEAlignmentOptions.SetFixedCoords* method). The default value is `0`, i.e. no atom clashes are allowed. If the number of clashes exceeds this limit the alignment is rejected and process falls back to maximizing the overlap between the *fit* and the *reference* 2D molecules by rotating around single bonds.

See also

- *OEAlignmentOptions.SetMaxAllowedAtomPairClashes* method

SetMaxBondRotations

```
void SetMaxBondRotations(unsigned int maxrotations)
```

Sets the maximum number of bond rotations performed during the alignment process.

maxrotations
: The zero value means that there is no limit. Upon reaching this limit, the alignment process terminates and returns the best alignment identified up to that point.

Hint

The default value for maximum number of bond rotations is $2^{16} = 65536$. This means that by default only the first 16 single bonds of the *fit* molecule will be rotated to maximize molecule alignment.

It is recommended to use this parameter in the following cases:

- if the *fit* molecule being aligned has a large number of rotatable bonds
- if the match that determines the atoms of the *fit* molecule that will be aligned to the atoms of the *reference* molecule has a large number of rotatable bonds

See also

- *OEAlignmentOptions.GetMaxBondRotations* method

SetMaxPatternMatches

```
void SetMaxPatternMatches(unsigned int maxmatches)
```

Sets the maximum number of matches utilized to perform molecule alignments.

maxmatches
: The zero value means that there is no limit. If the molecule alignment is based on more than one matches, then

upon reaching this limit, the alignment process terminates and returns the best alignment identified up to that point.

Hint

It is recommended to use this parameter in the following cases:

- if the alignment is based on substructure or maximum common substructure matches and the *reference* or/and *fit* molecules are highly symmetrical *i.e.* a large number of matches returned by the substructure or maximum common substructure search algorithms, respectively.

See also

- *OEAlignmentOptions.GetMaxPatternMatches* method

SetRMSDCutoff

```
void SetRMSDCutoff(double rmsdcutoff)
```

Sets the RMSD cutoff value used to identify acceptable alignments.

rmsdcutoff
This number has to be non-negative number. The alignment procedure terminates when the *rmsdcutoff* between the *fit* and the *reference* molecule is below this limit and there is no internal clash detected between the atoms of the *fit* molecule.

See also

- *OEAlignmentOptions.GetRMSDCutoff* method

SetRotateAroundBonds

```
void SetRotateAroundBonds(bool rotatebonds)
```

Sets whether rotation around the bonds of the *fit* molecule is allowed.

rotatebonds
If 'true', then bond of the *fit* molecule will be rotated to find the best alignment between molecules. If 'false', then only 2D rotation and translation of the *fit* molecule is allowed (*i.e.* no rotations around bonds) in order to align it to the *reference* molecule

See also

- *OEAlignmentOptions.GetRotateAroundBonds* method

SetSuppressHydrogens

```
void SetSuppressHydrogens(bool)
```

Sets whether the explicit hydrogens are suppressed in the *fit* molecule prior to the alignment process. Only hydrogens that are necessary to faithfully represent tetrahedral stereochemistry will be kept.

See also

- *OEAddDepictionHydrogens* function
- *OEHasDepictionHydrogens* function
- *OEAlignmentOptions.GetSuppressHydrogens* method

4.1.9 OEAlignmentResult

```
class OEAlignmentResult : public OEChem::OEMatchBase
```

The *OEAlignmentResult* class is a concrete instance of the OEMatchBase abstract base class. An *OEAlignmentResult* object represents a pairwise correspondence between atoms and/or bonds in two different molecules being aligned by the *OEPrepareAlignedDepiction* function.

The following methods are publicly inherited from OEMatchBase:

operator bool	GetBonds	GetTargetBonds
Clear	GetPatternAtoms	IsValid
CreateCopy	GetPatternBonds	NumAtoms
GetAtoms	GetTargetAtoms	NumBonds

See also

- OEMatchBase class in the **OEChem TK** manual
- *OEPrepareAlignedDepiction* function
- *Molecule Alignment* chapter

Constructors

```
OEAlignmentResult()
```

Default constructor.

```
OEAlignmentResult(const OEChem::OEMatchBase& match)
```

Constructors an *OEAlignmentResult* object from a match.

```
OEAlignmentResult(const OEAlignmentResult& rhs)
```

Copy constructor.

Clear

```
void Clear()
```

Removes all atom and bond of pairs from the *OEAlignmentResult* object.

CreateCopy

```
OEMatchBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned object derived from OEMatchBase is dynamically allocated and owned by the caller.

GetAtoms

```
OESystem::OEIterBase<OEChem::OEMatchPair<OEChem::OEAtomBase> >* GetAtoms() const
```

Returns an iterator over OEMatchPair which contain the atom to atom correspondences in the *OEAlignmentResult* object.

See also

OEMatchPair class in the **OEChem TK** manual

GetBonds

```
OESystem::OEIterBase<OEChem::OEMatchPair<OEChem::OEBondBase> >* GetBonds() const
```

Returns an iterator over OEMatchPair which contain the bond to bond correspondences in the *OEAlignmentResult* object.

See also

OEMatchPair class in the **OEChem TK** manual

GetRMSD

```
double GetRMSD() const
```

Returns the RMSD of the alignment based on the correspondence stored in the *OEAlignmentResult* object.

IsValid

```
bool IsValid() const
```

Returns true if there at least one atom to atom or bond to bonds correspondence is stored in the *OEAlignmentResult* object *i.e.* whether the alignment performed by the *OEAlignmentResult* function was successful.

NumAtoms

```
unsigned int NumAtoms() const
```

Returns the number of atom equivalences in the *OEAlignmentResult* object.

NumBonds

```
unsigned int NumBonds() const
```

Returns the number of bond equivalences in the *OEAlignmentResult* object.

4.1.10 OEAtomDisplayBase

```
class OEAtomDisplayBase : public OESystem::OEBase
```

The *OEAtomDisplayBase* class is the abstract interface for representing atom display information within **OEDepict TK**. See *Figure: OEDepict TK atom display class hierarchy*.

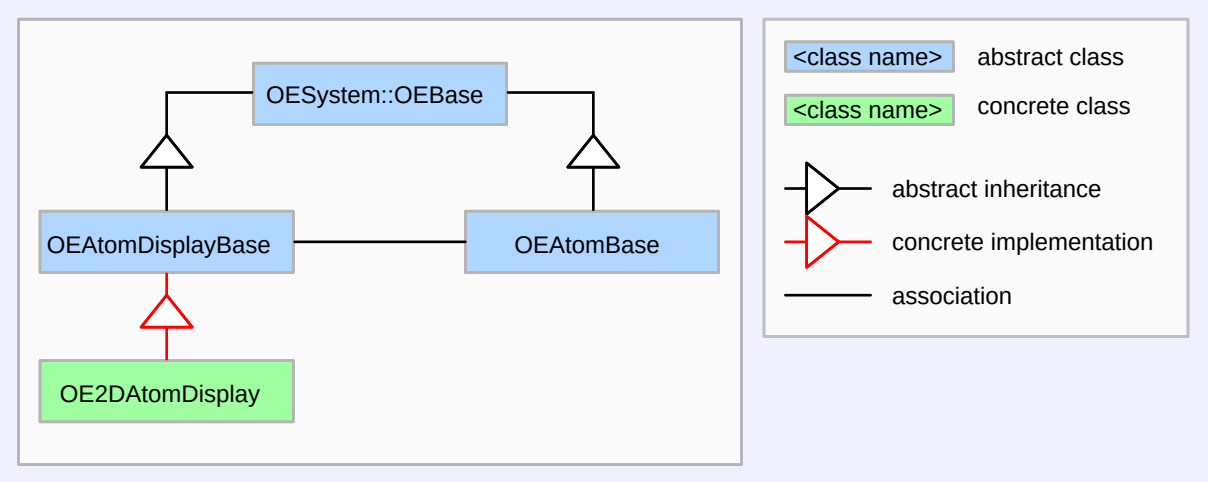


Fig. 69: **OEDepict TK atom display class hierarchy**

The following classes derive from this class:

- *OE2DAtomDisplay*

GetDataType

```
const void* GetDataType() const
```

This function is used to perform run-time type identification.

See also

- `OEBase.GetDataType` method in the **OEChem TK** manual

GetAtom

```
const OEChem::OEAtomBase* GetAtom() const
```

Returns the const pointer of the OEAtomBase object for which display properties are stored in the class derived from the *OEAtomDisplayBase* abstract class.

IsDataType

```
bool IsDataType(const void *) const
```

Returns whether type is the same as the instance this method is called on.

See also

- `OEBase.IsDataType` method in the **OEChem TK** manual

IsVisible

```
bool IsVisible() const
```

Returns whether or not the atom is depicted (*i.e.* visible).

SetVisible

```
void SetVisible(bool visible)
```

Sets the visibility of an atom.

4.1.11 OEAtomSVGAtomIdxMarkup

```
class OEAtomSVGAtomIdxMarkup : public OEAtomSVGMarkupBase
```

This class represents the *OEAtomSVGAtomIdxMarkup* atom SVG markup functor. If this functor is passed to the *`OE2DMolDisplayOptions.SetAtomSVGMarkupFunctor`* method, the atoms will be not marked in the `svg` image when calling the *OERenderMolecule* function.

See also

- *Generating Interactive SVG Images* chapter
- *`OE2DMolDisplayOptions.SetAtomSVGMarkupFunctor`* method
- *OEAtomSVGNoMarkup* class
- *OEAtomSVGResidueMarkup* class

The following methods are publicly inherited from *OEAtomSVGMarkupBase*:

CreateCopy	*GetClassName*	*GetGroupId*

Constructors

```
OEAtomSVGAtomIdxMarkup(const std::string prefix = "",
                       const std::string classname = "atom")
```

Creates an *OEAtomSVGAtomIdxMarkup* object.

prefix
: The string that will be added at the beginning of each atom *group id*. For information about valid ids see *OEIsValidSVGGroupId* function. By default, no prefix will be added.

classname
: The *class name* that is going to be used to mark atoms. For information about valid names see *OEIsValidSVGClassName* function.

By default, the svg image for the first atom of a molecule will be marked as follows:

```
<g id='idx=0' class='atom'>
..
list of drawing elements
..
<circle ... >
</g>
```

The transparent circle will be drawn with the atom position *OE2DAtomDisplay.GetCoords* center. This can be used to manipulate the generated `.svg` image and associate event with the SVG groups.

CreateCopy

```
OEAtomSVGMarkupBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEAtomSVGAtomIdxMarkup* object is dynamically allocated and owned by the caller.

GetClassName

```
std::string GetClassName(const OEChem::OEAtomBase &) const
```

Returns the string that specifies the *group id* of an atom in an `svg` image.

GetGroupId

```
std::string GetGroupId(const OEChem::OEAtomBase &) const
```

Returns the string that specifies the *class name* of an atom in an `svg` image.

4.1.12 OEAtomSVGMarkupBase

```
class OEAtomSVGMarkupBase
```

OEAtomSVGMarkupBase is an abstract base class. Concrete classes derived from the base class define the markup of the atoms in `.svg` images that are generated when calling the *OERenderMolecule* function. Drawing elements representing atoms in the `.svg` images are grouped together in the following format in which the `<group id>` and `<class name>` strings can be defined in the derived concrete classes.

```
<g id='<group id>' class='<class name>'>
..
list of drawing elements
..
</g>
```

The following classes derive from this class:

- *OEAtomSVGAtomIdxMarkup*
- *OEAtomSVGNoMarkup*
- *OEAtomSVGResidueMarkup*

See also

- *OE2DMolDisplayOptions.SetAtomSVGMarkupFunctor* method
- *Generating Interactive SVG Images* chapter

Note

User defined SVG atom marking can be implemented by deriving from this class and implementing all the methods listed below.

CreateCopy

```
OEAtomSVGMarkupBase* CreateCopy() const = 0
```

Virtual const constructor which allows copying of concrete derived objects using a reference to this base class.

GetClassName

```
std::string GetClassName(const OEChem::OEAtomBase &) const = 0
```

Returns the <class name> associated with a specific atom group in the .svg image. This is a virtual method that has to be implemented in the concrete derived classes. The return string will be verified by the *OEIsValidSVGClassName* function. Invalid SVG class names will be ignored.

See also

- *OEIsValidSVGClassName* function

GetGroupId

```
std::string GetGroupId(const OEChem::OEAtomBase &) const = 0
```

Returns the <group id> associated with a specific atom group in the .svg image. This is a virtual method that has to be implemented in the concrete derived classes. The return string will be verified by the *OEIsValidSVGGroupId* function. Invalid SVG group ids will be ignored.

See also

- *OEIsValidSVGGroupId* function

4.1.13 OEAtomSVGNoMarkup

```
class OEAtomSVGNoMarkup : public OEAtomSVGMarkupBase
```

This class represents the *OEAtomSVGNoMarkup* atom SVG markup functor. If this functor is passed to the *OE2DMolDisplayOptions.SetAtomSVGMarkupFunctor* method, the atoms will be not marked in the svg image when calling the *OERenderMolecule* function.

See also

- *Generating Interactive SVG Images* chapter
- *OE2DMolDisplayOptions.SetAtomSVGMarkupFunctor* method
- *OEAtomSVGAtomIdxMarkup* class
- *OEAtomSVGResidueMarkup* class

The following methods are publicly inherited from *OEAtomSVGMarkupBase*:

CreateCopy	*GetClassName*	*GetGroupId*

Constructors

```
OEAtomSVGNoMarkup()
```

Default constructor.

CreateCopy

```
OEAtomSVGMarkupBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEAtomSVGNoMarkup* object is dynamically allocated and owned by the caller.

GetClassName

```
std::string GetClassName(const OEChem::OEAtomBase &) const
```

Returns an empty string.

GetGroupId

```
std::string GetGroupId(const OEChem::OEAtomBase &) const
```

Returns an empty string.

4.1.14 OEAtomSVGResidueMarkup

```
class OEAtomSVGResidueMarkup : public OEAtomSVGMarkupBase
```

This class represents the *OEAtomSVGResidueMarkup* atom SVG markup functor. If this functor is passed to the *OE2DMolDisplayOptions.SetAtomSVGMarkupFunctor* method, the atoms will be not marked in the `svg` image when calling the *OERenderMolecule* function.

See also

- *Generating Interactive SVG Images* chapter
- *OE2DMolDisplayOptions.SetAtomSVGMarkupFunctor* method
- *OEAtomSVGAtomIdxMarkup* class
- *OEAtomSVGNoMarkup* class

The following methods are publicly inherited from *OEAtomSVGMarkupBase*:

CreateCopy	*GetClassName*	*GetGroupId*

Constructors

```
OEAtomSVGResidueMarkup(const std::string prefix = "",
                       const std::string classname = "ligand-atom",
                       const char separator = '-')
```

Creates an *OEAtomSVGResidueMarkup* object.

prefix
: The string that will be added at the beginning of each atom *group id*. For information about valid ids see *OEIsValidSVGGroupId* function. By default, no prefix will be added.

classname
: The *class name* that is going to be used to mark atoms. For information about valid names see *OEIsValidSVGClassName* function.

separator
: The character used to separate the components (residue name, chain id, residue number, atom name) of the *group id*

By default, the following will be showed in `svg` image for marking of the first atom of molecule `NC(CCCNC(N)=N)C(O)=O` with residue information:

By default, the `svg` image for the first atom of molecule `NC(CCCNC(N)=N)C(O)=O` with residue information will be marked as follows:

```
<g id='ARG-A-1-_N__' class='ligand-atom'>
..
list of drawing elements
..
<circle ... >
</g>
```

If there is no residue information attached to the atom the UNK label will be used for the residue name. The residue information can be perceived by calling the OEPerceiveResidues function.

Since a valid *group id* can not have a space character, each space character of the atom name (returned by OEAtomBase.GetName) is replaces by an _ (underline).

The transparent circle will be drawn with the atom position *OE2DAtomDisplay.GetCoords* center. This can be used to manipulate the generated .svg image and associate event with the SVG groups.

CreateCopy

```
OEAtomSVGMarkupBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEAtomSVGResidueMarkup* object is dynamically allocated and owned by the caller.

GetClassName

```
std::string GetClassName(const OEChem::OEAtomBase &) const
```

Returns the string that specifies the *group id* of an atom in an svg image.

GetGroupId

```
std::string GetGroupId(const OEChem::OEAtomBase &) const
```

Returns the string that specifies the *class name* of an atom in an svg image.

4.1.15 OEBondDisplayBase

```
class OEBondDisplayBase : public OESystem::OEBase
```

The *OEBondDisplayBase* class is the abstract interface for representing bond display information within **OEDepict TK**. See *Figure: OEDepict TK bond display class hierarchy*.

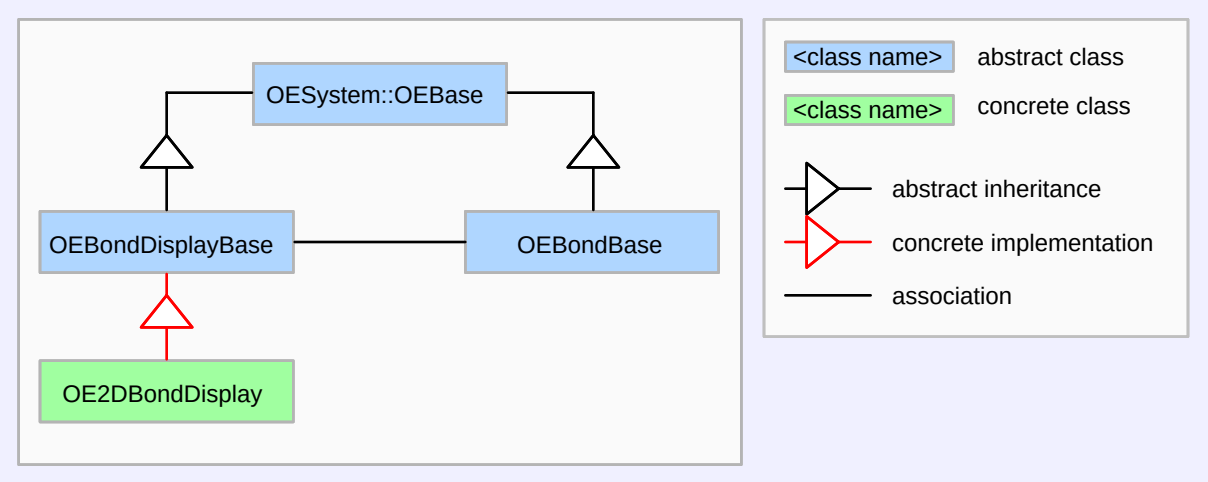


Fig. 70: **OEDepict TK bond display class hierarchy**

The following classes derive from this class:

- *OE2DBondDisplay*

GetDataType

```
const void* GetDataType() const
```

This function is used to perform run-time type identification.

See also

- `OEBase.GetDataType` method in the **OEChem TK** manual

GetBond

```
const OEChem::OEBondBase* GetBond() const
```

Returns the const pointer of the OEBondBase object for which display properties are stored in the class derived from the *OEBondDisplayBase* abstract class.

IsDataType

```
bool IsDataType(const void *) const
```

Returns whether type is the same as the instance this method is called on.

See also

- `OEBase.IsDataType` method in the **OEChem TK** manual

IsVisible

```
bool IsVisible() const
```

Returns whether or not the bond is depicted (*i.e.* visible).

SetVisible

```
void SetVisible(bool visible)
```

Sets the visibility of a bond.

4.1.16 OEBondSVGBondIdxMarkup

```
class OEBondSVGBondIdxMarkup : public OEBondSVGMarkupBase
```

This class represents the *OEBondSVGBondIdxMarkup* bond SVG markup functor. If this functor is passed to the *`OE2DMolDisplayOptions.SetBondSVGMarkupFunctor`* method, the bonds will be not marked in the `svg` image when calling the *OERenderMolecule* function.

See also

- *Generating Interactive SVG Images* chapter
- *OE2DMolDisplayOptions.SetBondSVGMarkupFunctor* method
- *OEBondSVGNoMarkup* class

The following methods are publicly inherited from *OEBondSVGMarkupBase*:

CreateCopy	*GetClassName*	*GetGroupId*

Constructors

```
OEBondSVGBondIdxMarkup(const std::string prefix = "",
                       const std::string classname = "bond")
```

Creates an *OEBondSVGBondIdxMarkup* object.

prefix
: The string that will be added at the beginning of each bond *group id*. For information about valid ids see *OEIsValidSVGGroupId* function. By default, no prefix will be added.

classname
: The *class name* that is going to be used to mark bonds. For information about valid names see *OEIsValidSVGClassName* function.

By default, the svg image for the first bond of a molecule will be marked as follows:

```
<g id='idx=0' class='bond'>
..
list of drawing elements
..
<circle ... >
</g>
```

The transparent circle will be drawn at the middle of the bond. This can be used to manipulate the generated `.svg` image and associate event with the SVG groups.

CreateCopy

```
OEBondSVGMarkupBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEBondSVGBondIdxMarkup* object is dynamically allocated and owned by the caller.

GetClassName

```
std::string GetClassName(const OEChem::OEBondBase &) const
```

Returns the string that specifies the *group id* of an bond in an `svg` image.

GetGroupId

```
std::string GetGroupId(const OEChem::OEBondBase &) const
```

Returns the string that specifies the *class name* of an bond in an svg image.

4.1.17 OEBondSVGMarkupBase

```
class OEBondSVGMarkupBase
```

OEBondSVGMarkupBase is an abstract base class. Concrete classes derived from the base class define the markup of the bonds in .svg images that are generated when calling the *OERenderMolecule* function. Drawing elements representing bonds in the .svg images are grouped together in the following format in which the <group id> and <class name> strings can be defined in the derived concrete classes.

```
<g id='<group id>' class='<class name>'>
..
list of drawing elements
..
</g>
```

The following classes derive from this class:

- *OEBondSVGBondIdxMarkup*
- *OEBondSVGNoMarkup*

See also

- *OE2DMolDisplayOptions.SetBondSVGMarkupFunctor* method
- *Generating Interactive SVG Images* chapter

Note

User defined SVG atom marking can be implemented by deriving from this class and implementing all the methods listed below.

CreateCopy

```
OEBondSVGMarkupBase* CreateCopy() const = 0
```

Virtual const constructor which allows copying of concrete derived objects using a reference to this base class.

GetClassName

```
std::string GetClassName(const OEChem::OEBondBase &) const = 0
```

Returns the <class name> associated with a specific bond group in the .svg image. This is a virtual method that has to be implemented in the concrete derived classes. The return string will be verified by the *OEIsValidSVGClassName* function. Invalid SVG class names will be ignored.

See also

- *OEIsValidSVGClassName* function

GetGroupId

```
std::string GetGroupId(const OEChem::OEBondBase &) const = 0
```

Returns the <group id> associated with a specific bond group in the .svg image. This is a virtual method that has to be implemented in the concrete derived classes. The return string will be verified by the *OEIsValidSVGGroupId* function. Invalid SVG group ids will be ignored.

See also

- *OEIsValidSVGGroupId* function

4.1.18 OEBondSVGNoMarkup

```
class OEBondSVGNoMarkup : public OEBondSVGMarkupBase
```

This class represents the *OEBondSVGNoMarkup* bond SVG markup functor. If this functor is passed to the *OE2DMolDisplayOptions.SetBondSVGMarkupFunctor* method, the bonds will be not marked in the svg image when calling the *OERenderMolecule* function.

See also

- *Generating Interactive SVG Images* chapter
- *OE2DMolDisplayOptions.SetBondSVGMarkupFunctor* method
- *OEBondSVGBondIdxMarkup* class

The following methods are publicly inherited from *OEBondSVGMarkupBase*:

CreateCopy	*GetClassName*	*GetGroupId*

Constructors

```
OEBondSVGNoMarkup()
```

Default constructor.

CreateCopy

```
OEBondSVGMarkupBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEBondSVGNoMarkup* object is dynamically allocated and owned by the caller.

GetClassName

```
std::string GetClassName(const OEChem::OEBondBase &) const
```

Returns an empty string.

GetGroupId

```
std::string GetGroupId(const OEChem::OEBondBase &) const
```

Returns an empty string.

4.1.19 OEDisplayAtomIdx

```
class OEDisplayAtomIdx : public OEDisplayAtomPropBase
```

This class represents *OEDisplayAtomIdx* atom property label functor.

See also

- *OE2DMolDisplayOptions.SetAtomPropertyFunctor* method
- *OEDisplayAtomMapIdx* class
- *OEDisplayNoAtomProp* class
- *Listing 5* example in the *Molecule Depiction* chapter

The following methods are publicly inherited from *OEDisplayAtomPropBase*:

CreateCopy	*operator()*

Constructors

```
OEDisplayAtomIdx()
```

Default constructor.

operator()

```
std::string operator()(const OEChem::OEAtomBase& atom) const
```

Returns the string representation of the index of the atom (*i.e.* the number returned by the `OEAtomBase.GetIdx` method).

CreateCopy

```
base_type* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEDisplayAtomIdx* object is dynamically allocated and owned by the caller.

4.1.20 OEDisplayAtomMapIdx

```
class OEDisplayAtomMapIdx : public OEDisplayAtomPropBase
```

This class represents *OEDisplayAtomMapIdx* atom property label functor.

See also

- *OE2DMolDisplayOptions.SetAtomPropertyFunctor* method
- *OEDisplayAtomIdx* class
- *OEDisplayNoAtomProp* class
- *Listing 2* example in the *MDL Reaction Depiction* chapter

The following methods are publicly inherited from *OEDisplayAtomPropBase*:

CreateCopy	*operator()*

Constructors

```
OEDisplayAtomMapIdx()
```

Default constructor.

operator()

```
std::string operator()(const OEChem::OEAtomBase& atom) const
```

Returns the string representation of the *reaction map index* property of the atom. (i.e. the number returned by the `OEAtomBase.GetMapIdx` method). The *reaction map index* is used for tracking equivalent atom positions in the reaction. If *the reaction map index* is zero, meaning that the atom isn't mapped in the reaction, than this function returns an empty string.

CreateCopy

```
base_type* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEDisplayAtomMapIdx* object is dynamically allocated and owned by the caller.

4.1.21 OEDisplayAtomPropBase

```
OESystem::OEUnaryFunction<OEChem::OEAtomBase, std::string, true, true>
OEDisplayAtomPropBase
```

OEDisplayAtomPropBase is an abstract class that defines the API used for setting atom property labels.

See also

- *OE2DMolDisplayOptions.SetAtomPropertyFunctor* method

- *Listing 6* example in the *Molecule Depiction* chapter

The following classes derive from this class:

- *OEDisplayAtomIdx* class
- *OEDisplayAtomMapIdx* class
- *OEDisplayNoAtomProp* class

operator()

```
std::string operator () (const OEChem::OEAtomBase& atom) const
```

It is a virtual const method that is implemented in the concrete derived classes to set atom property labels.

atom
The OEAtomBase object for which the display label is set.

CreateCopy

```
base_type* CreateCopy() const = 0
```

Virtual const constructor which allows copying of concrete derived objects using a reference to this base class.

4.1.22 OEDisplayBondIdx

```
class OEDisplayBondIdx : public OEDisplayBondPropBase
```

This class represents *OEDisplayBondIdx*.

See also

- *OE2DMolDisplayOptions.SetBondPropertyFunctor* method
- *OEDisplayNoBondProp* class
- *Listing 7* example in the *Molecule Depiction* chapter

The following methods are publicly inherited from *OEDisplayBondPropBase*:

CreateCopy *operator()*

Constructors

```
OEDisplayBondIdx()
```

Default constructor.

operator()

```
std::string operator()(const OEChem::OEBondBase& bond) const
```

Returns the string representation of the index of the bond (*i.e.* the number returned by the `OEBondBase.GetIdx` method)

CreateCopy

```
base_type* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEDisplayBondIdx* object is dynamically allocated and owned by the caller.

4.1.23 OEDisplayBondPropBase

```
OESystem::OEUnaryFunction<OEChem::OEBondBase, std::string, true, true>
class OEDisplayBondPropBase
```

OEDisplayBondPropBase is an abstract class that defines the API used for setting bond property labels.

See also

- *OE2DMolDisplayOptions.SetBondPropertyFunctor* method
- *Listing 8* example in the *Molecule Depiction* chapter

The following classes derive from this class:

- *OEDisplayBondIdx* class
- *OEDisplayNoBondProp* class

operator()

```
std::string operator () (const OEChem::OEBondBase& bond) const
```

It is a virtual const method that is implemented in the concrete derived classes to set bond property labels.

bond
: The OEBondBase object for which the display label is set.

CreateCopy

```
base_type* CreateCopy() const = 0
```

Virtual const constructor which allows copying of concrete derived objects using a reference to this base class.

4.1.24 OEDisplayNoAtomProp

```
class OEDisplayNoAtomProp : public OEDisplayAtomPropBase
```

This class represents *OEDisplayNoAtomProp* atom property label functor.

See also

- *OE2DMolDisplayOptions.SetAtomPropertyFunctor* method
- *OEDisplayAtomIdx* class
- *OEDisplayAtomMapIdx* class

The following methods are publicly inherited from *OEDisplayAtomPropBase*:

CreateCopy	*operator()*

Constructors

```
OEDisplayNoAtomProp()
```

Default constructor.

operator()

```
std::string operator()(const OEChem::OEAtomBase & atom) const
```

Returns an empty string which means no atom property is displayed.

CreateCopy

```
base_type* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEDisplayNoAtomProp* object is dynamically allocated and owned by the caller.

4.1.25 OEDisplayNoBondProp

```
class OEDisplayNoBondProp : public OEDisplayBondPropBase
```

This class represents *OEDisplayNoBondProp*.

The following methods are publicly inherited from *OEDisplayBondPropBase*:

CreateCopy	*operator()*

See also

- *OE2DMolDisplayOptions.SetBondPropertyFunctor* method
- *OEDisplayBondIdx* class

Constructors

```
OEDisplayNoBondProp()
```

Default constructor.

operator()

```
std::string operator()(const OEChem::OEBondBase & bond) const
```

Returns an empty string, which means that no bond property is displayed.

CreateCopy

```
base_type* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEDisplayNoBondProp* object is dynamically allocated and owned by the caller.

4.1.26 OEFont

```
class OEFont
```

The *OEFont* class encapsulates properties that determine the display style of texts.

Currently the *OEFont* class stores the following properties:

Property	Get method	Set method	Corresponding namespace/class/type
alignment	*GetAlignment*	*SetAlignment*	*OEAlignment*
color	*GetColor*	*SetColor*	OEColor
font family	*GetFamily*	*SetFamily*	*OEFontFamily*
font size	*GetSize*	*SetSize*	None
font style	*GetStyle*	*SetStyle*	*OEFontStyle*
hyperlink	*GetHyperlink* / *HasHyperlink*	*SetHyperlink*	string
rotation angle	*GetRotationAngle*	*SetRotationAngle*	floating point number in range of $[0.0°, 360.0°]$

See also

- *OEImageBase.DrawText* method

Constructors

```
OEFont()
```

Default constructor that initializes an *OEFont* object with the following properties: See example in *Figure: Example of drawing text with default OEFont object*.

Table 6: Default parameters of OEFont

Property	Default value
alignment	*OEAlignment.Default*
color	OEBlack = OEColor(0,0,0)
font family	*OEFontFamily.Default*
font size	12
hyperlink	empty string (i.e. no hyperlink)
font style	*OEFontStyle.Default*
rotation angles	0.0 (i.e. no rotation)



Fig. 71: **Example of drawing text with default OEFont object**

```
OEFont(const OESystem::OEColor& color)
```

Constructor that initializes an *OEFont* object with the default parameters and the given OEColor.

```
OEFont(unsigned int family, unsigned int style, unsigned int size,
       unsigned int align, const OESystem::OEColor& color,
       double rotangle = 0.0)
```

Constructor that initializes an *OEFont* object with the given parameters.

family
: The family of the font. This value has to be from the *OEFontFamily* namespace. See also *OEFont.SetFamily* method.

style
: The style of the font. This value has to be from the *OEFontStyle* namespace. See also *OEFont.SetSize* method.

size
: The size of the font. See also *OEFont.SetSize* method.

align
: The alignment of the text. This value has to be from the *OEAlignment* namespace. See also *OEFont.SetAlignment* method.

color
: The color of the font. See also *OEFont.SetColor* method.

rotangle
: The rotation angle of the text. The default value is 0.0 which means no rotation is applied *i.e.* the text using the *OEFont* object is rendered horizontally. See also *OEFont.SetRotationAngle* method.

```
OEFont(const OEFont& rhs)
```

Copy constructors.

GetAlignment

```
unsigned int GetAlignment() const
```

Returns the *alignment* property of the *OEFont* object. The returned value is taken from the *OEAlignment* namespace.

See also

- *OEFont.SetAlignment* method
- *OEAlignment* namespace

GetColor

```
const OESystem::OEColor& GetColor() const
```

Returns the color of the *OEFont* object.

See also

- *OEFont.SetColor* method
- OEColor class

GetFamily

```
unsigned int GetFamily() const
```

Returns the *font family* property of the *OEFont* object. The returned value is taken from the *OEFontFamily* namespace.

See also

- *OEFont.SetFamily* method
- *OEFontFamily* namespace

GetHyperlink

```
const std::string& GetHyperlink() const
```

Returns the URI string associated with the *OEFont* object.

See also

- *OEFont.HasHyperlink* method
- *OEFont.SetHyperlink* method

GetRotationAngle

```
double GetRotationAngle() const
```

Returns the angle of the rotation that is used to render a text when using the *OEFont* object.

See also

- *OEFont.SetRotationAngle* method

GetSize

```
unsigned int GetSize() const
```

Returns the size of the *OEFont* object.

See also

- *OEFont.SetSize* method

GetStyle

```
unsigned int GetStyle() const
```

Returns the *font style* property of the *OEFont* object. The returned value is taken from the *OEFontStyle* namespace.

See also

- *OEFont.SetStyle* method
- *OEFontStyle* namespace

HasHyperlink

```
bool HasHyperlink() const
```

Returns whether an URI string is associated with the *OEFont* object.

See also

- *OEFont.GetHyperlink* method
- *OEFont.SetHyperlink* method

SetAlignment

```
void SetAlignment(unsigned int align)
```

Sets the *alignment* property of the *OEFont* object.

align
: This value has to be from the *OEAlignment* namespace.

Example (*Figure: Example of using the SetAlignment method*)

```
OEFont font = new OEFont();
font.SetAlignment(OEAlignment.Right);
```



Fig. 72: **Example of using the SetAlignment method**

See also

- *OEFont.GetAlignment* method
- *OEAlignment* namespace

SetColor

```
void SetColor(const OESystem::OEColor& color)
```

Sets the color of the *OEFont* object.

Example (*Figure: Example of using the SetColor method*)

```
OEFont font = new OEFont();
font.SetColor(oechem.getOERed());
```



Fig. 73: **Example of using the SetColor method**

See also

- *OEFont.GetColor* method
- OEColor class

SetFamily

```
void SetFamily(unsigned int family)
```

Sets the *font family* property of the *OEFont* object.

family
This value has to be from the *OEFontFamily* namespace.

Example (*Figure: Example of using the SetFamily method*)

```
OEFont font = new OEFont();
font.SetFamily(OEFontFamily.Times);
```



Fig. 74: **Example of using the SetFamily method**

See also

- *OEFont.GetFamily* method
- *OEFontFamily* namespace

SetHyperlink

```
void SetHyperlink(const string& uri)
```

Associates URI string with the *OEFont* object.

uri
After associating a **valid** URI with a *OEFont* object, texts drawn with the given font become "click-able" and by clicking on them will send you to the given address.

Example (*Figure: Example of using the SetHyperlink method*)

```
OEGraphMol mol = new OEGraphMol();
oechem.OESmilesToMol(mol, "OC(=O)c1ccccc1");
oedepict.OEPrepareDepiction(mol);

double width = 150.0;
double height = 150.0;
```

(continues on next page)

(continued from previous page)

```
double scale =  OEScale.AutoScale;

OEImage image = new OEImage(width, height);
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);
opts.SetTitleLocation(OETitleLocation.Bottom);

// associate font with URI

OEFont font = new OEFont(OEFontFamily.Default, OEFontStyle.Default, 10,
                         OEAlignment.Center, oechem.getOEBlue());
font.SetHyperlink("http://www.eyesopen.com/oedepict-tk");

opts.SetTitleFont(font);
mol.SetTitle("Generated by OEDepict");

OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
oedepict.OERenderMolecule(image, disp);

oedepict.OEWriteImage("DepictMolWithHyperlink.pdf", image);
oedepict.OEWriteImage("DepictMolWithHyperlink.svg", image);
```

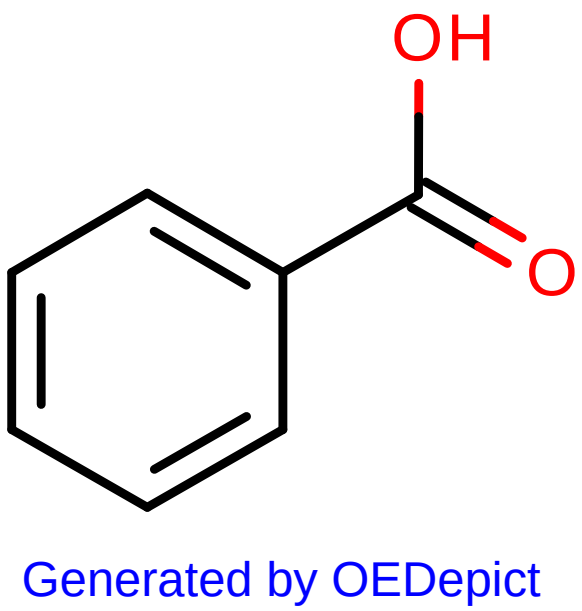


Fig. 75: **Example of using the SetHyperlink method**
(Clicking on the blue text at the bottom send you to the OEDepict TK web page)

Note

Associating texts with hyperlinks is **only available** for `.pdf` and `.svg` image formats.

Warning

OEDepict TK does **not** check whether the given string is a valid URI.

It is recommended to always keep the URI scheme prefix of the URI string for the sake of compatibility. For example, some application can interpret the `www.eyesopen.com` string as a valid URI, others only work when using `http://www.eyesopen.com`.

See also

- *OEFont.GetHyperlink* method
- *OEFont.GetHyperlink* method

SetRotationAngle

```
void SetRotationAngle(double angle)
```

Sets the rotation angle of the *OEFont* object.

angle
The rotation angle of the text (in degrees). The value of the angle has to be in a range of $[0.0°, 360.0°]$.

Example (*Figure: Example of using the SetRotationAngle method*)

```
OEFont font = new OEFont();
font.SetRotationAngle(180.0);
```

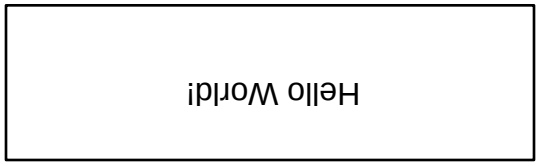


Fig. 76: **Example of using the SetRotationAngle method**

Note

The angles are interpreted in an anti-clockwise direction. The 0.0° and 360.0° degrees are at the 3 o'clock position *i.e.* the text is rendered horizontally. While the 180.0° degrees corresponds to rendering the text upside down. See examples in *Figure: Example of drawing texts with various rotation angle*.



Fig. 77: **Example of drawing texts with various rotation angles**

See also

- *OEFont.GetRotationAngle* method

SetSize

```
void SetSize(unsigned int size)
```

Sets the size of the *OEFont* object.

Example (*Figure: Example of using the SetSize method*)

```
OEFont font = new OEFont();
font.SetSize(30);
```



Fig. 78: **Example of using the SetSize method**

See also

- *OEFont.GetSize* method

SetStyle

```
void SetStyle(unsigned int style)
```

Sets the *font style* property (such as **bold**, *italic* etc.) of the *OEFont* object.

style
: This value has to be from the *OEFontStyle* namespace.

Example (*Figure: Example of using the SetStyle method*)

```
OEFont font = new OEFont();
font.SetStyle(OEFontStyle.Bold|OEFontStyle.Italic);
```



Fig. 79: **Example of using the SetStyle method**

See also

- *OEFont.GetStyle* method
- *OEFontStyle* namespace

4.1.27 OEHighlightBase

```
class OEHighlightBase
```

OEHighlightBase is an abstract class used for highlighting atoms and/or bonds.

See also

- *OEAddHighlighting* function.

The following classes derive from this class:

- *OEHighlightByBallAndStick*
- *OEHighlightByColor*
- *OEHighlightByCogwheel*
- *OEHighlightByLasso*
- *OEHighlightByStick*

Constructors

```
OEHighlightBase()
OEHighlightBase(OEHighlightBaseImpl* impl)
```

Default constructors.

CreateCopy

```
virtual OEHighlightBase* CreateCopy() const = 0
```

Virtual const constructor which allows copying of concrete derived objects using a reference to this base class.

4.1.28 OEHighlightByBallAndStick

```
class OEHighlightByBallAndStick : public OEHighlightBase
```

The *OEHighlightByBallAndStick* class allows the customization of the **ball and stick** highlighting style that is associated with the *OEHighlightStyle.BallAndStick* constant. See example in *Figure: Example of highlighting using the 'ball and stick' style*.

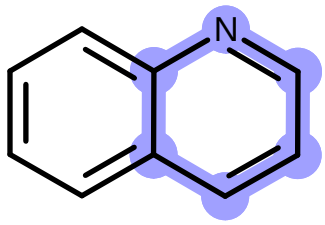


Fig. 80: **Example of highlighting using the 'ball and stick' style**

The *OEHighlightByBallAndStick* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
atom label	*GetAtomLabelMonochr*	*SetAtomLabelMonochr*	boolean
ball radius scale	*GetBallRadiusScale*	*SetBallRadiusScale*	floating point number
color	*GetColor*	*SetColor*	OEColor
stick width scale	*GetStickWidthScale*	*SetStickWidthScale*	floating point number

See also

- *OEAddHighlighting* function
- *OEHighlightStyle* namespace
- *OEHighlightByColor* class
- *OEHighlightByCogwheel* class

- *OEHighlightByLasso* class
- *OEHighlightByStick* class
- *Highlighting* chapter

Constructors

```
OEHighlightByBallAndStick(const OESystem::OEColor& c,
                          double stickWidthScale = 3.0,
                          double ballRadiusScale = 3.0,
                          bool monochrome = true)
```

Creates an *OEHighlightByBallAndStick* object with the specified parameters.

color
The color used for highlighting. See also *OEHighlightByBallAndStick.SetColor* method.

stickWidthScale
The multiplier that can be used to increase or decrease the stick width of the highlighted bond(s). See also *OEHighlightByBallAndStick.SetStickWidthScale* method.

ballRadiusScale
The multiplier that can be used to increase or decrease the ball radius of the highlighted atom(s). See also *OEHighlightByBallAndStick.SetBallRadiusScale* method.

monochrome
Defines whether or not to change the color of the atom label of the highlighted atom(s). See also *OEHighlightByBallAndStick.SetAtomLabelMonochrome* method.

```
OEHighlightByBallAndStick(const OEHighlightByBallAndStick& rhs)
```

Copy constructor.

CreateCopy

```
OEHighlightBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEHighlightByBallAndStick* object is dynamically allocated and owned by the caller.

GetAtomLabelMonochrome

```
bool GetAtomLabelMonochrome() const
```

Returns whether or not the color of the atom label of the highlighted atom(s) is changed.

See also

- *OEHighlightByBallAndStick.SetAtomLabelMonochrome* method

GetBallRadiusScale

```
double GetBallRadiusScale() const
```

Returns the multiplier that can be used to increase or decrease the ball radius of the highlighted atom(s).

See also

- *OEHighlightByBallAndStick.SetBallRadiusScale* method

GetColor

```
const OESystem::OEColor& GetColor() const
```

See also

- *OEHighlightByBallAndStick.SetColor* method
- OEColor class

GetStickWidthScale

```
double GetStickWidthScale() const
```

Returns the multiplier that can be used to increase or decrease the stick width of the highlighted bond(s).

See also

- *OEHighlightByBallAndStick.SetStickWidthScale* method

SetAtomLabelMonochrome

```
void SetAtomLabelMonochrome(bool monochrome)
```

Sets whether or not to change the color of the atom label of the highlighted atom(s).

Example (*Figure: Example of using the SetAtomLabelMonochrome method*)

```
OEHighlightByBallAndStick highlight = new OEHighlightByBallAndStick(oechem.
↪getOEPinkTint());
highlight.SetAtomLabelMonochrome(false);
```

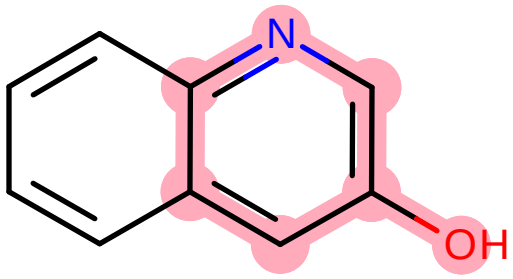


Fig. 81: **Example of using the SetAtomLabelMonochrome method**

See also

- *OEHighlightByBallAndStick.GetAtomLabelMonochrome* method

SetBallRadiusScale

```
void SetBallRadiusScale(double ballRadiusScale)
```

Sets the multiplier that can be used to increase or decrease the ball radius of the highlighted atom(s).

Example (*Figure: Example of using the SetBallRadiusScale method*)

```
OEHighlightByBallAndStick highlight = new OEHighlightByBallAndStick(oechem.
↪getOEPinkTint());
highlight.SetBallRadiusScale(4.0);
```

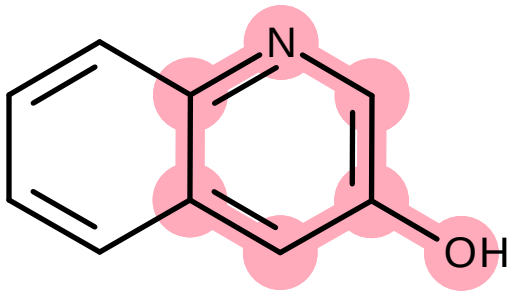


Fig. 82: **Example of using the SetBallRadiusScale method**

See also

- *OEHighlightByBallAndStick.GetBallRadiusScale* method

SetColor

```
void SetColor(const OESystem::OEColor& color)
```

Sets the color of the highlighting.

Example (*Figure: Example of using the SetColor method*)

```
highlight.SetColor(oechem.getOEPinkTint());
```

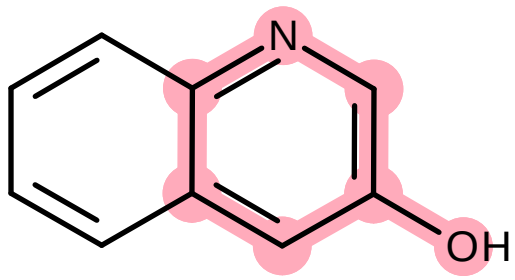


Fig. 83: **Example of using the SetColor method**

See also

- *OEHighlightByBallAndStick.GetColor* method
- OEColor class

SetStickWidthScale

```
void SetStickWidthScale(double stickWidthScale)
```

Sets the multiplier that can be used to increase or decrease the stick width of the highlighted bond(s).

Example (*Figure: Example of using the SetStickWidthScale method*)

```
OEHighlightByBallAndStick highlight = new OEHighlightByBallAndStick(oechem.
↪getOEPinkTint());
highlight.SetStickWidthScale(5.0);
```

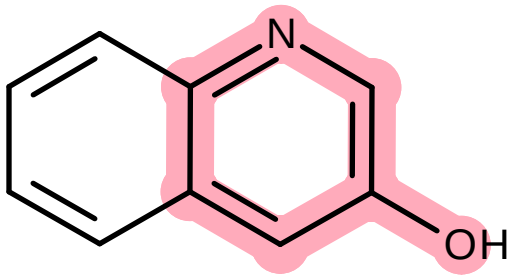


Fig. 84: **Example of using the SetStickWidthScale method**

See also

- *OEHighlightByBallAndStick.GetStickWidthScale* method

4.1.29 OEHighlightByCogwheel

```
class OEHighlightByCogwheel : public OEHighlightBase
```

The *OEHighlightByCogwheel* class allows the customization of the **cogwheel** highlighting style that is associated with the *OEHighlightStyle.Cogwheel* constant. See example in *Figure: Example of highlighting using the 'cogwheel' style*.

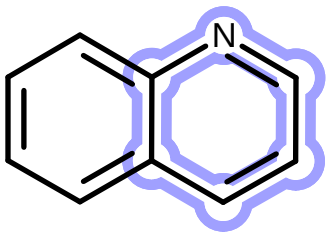


Fig. 85: **Example of highlighting using the color style**

The *OEHighlightByCogwheel* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
atom label	*GetAtomLabelMonochr*	*SetAtomLabelMonochr*	boolean
ball radius scale	*GetBallRadiusScale*	*SetBallRadiusScale*	floating point number
color	*GetColor*	*SetColor*	OEColor
inner contour	*GetInnerContour*	*SetInnerContour*	boolean
line width scale	*GetLineWidthScale*	*SetLineWidthScale*	floating point number
stick width scale	*GetStickWidthScale*	*SetStickWidthScale*	floating point number

See also

- *OEAddHighlighting* function
- *OEHighlightStyle* namespace
- *OEHighlightByBallAndStick* class
- *OEHighlightByColor* class
- *OEHighlightByLasso* class
- *OEHighlightByStick* class
- *Highlighting* chapter

Constructors

```
OEHighlightByCogwheel(const OESystem::OEColor& color,
                      double lineWidthScale = 1.5,
                      double stickWidthScale = 2.0,
                      double ballRadiusScale = 2.0,
                      bool innerContour = true,
                      bool monochrome = true)
```

Creates an *OEHighlightByCogwheel* object with the specified parameters.

color
: The color used for highlighting. See also *OEHighlightByCogwheel.SetColor* method.

lineWidthScale
: The multiplier that can be used to increase or decrease the line width of the 'cogwheel'. See also *OEHighlightByCogwheel.SetLineWidthScale* method.

stickWidthScale
: The multiplier that can be used to increase or decrease the stick width of the 'cogwheel' of the highlighted bond(s). See also *OEHighlightByCogwheel.SetStickWidthScale* method.

ballRadiusScale
: The multiplier that can be used to increase or decrease the ball radius of the 'cogwheel' of the highlighted atom(s). See also *OEHighlightByCogwheel.SetBallRadiusScale* method.

innerContour
: Defines whether the inner contour of the 'cogwheel' is drawn or committed. See also *OEHighlightByCogwheel.SetInnerContour* method.

monochrome
Defines whether or not to change the color of the atom label of the highlighted atom(s). See also *OEHighlightByCogwheel.SetAtomLabelMonochrome* method.

```
OEHighlightByCogwheel(const OEHighlightByCogwheel& rhs)
```

Copy constructor.

CreateCopy

```
OEHighlightBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEHighlightByCogwheel* object is dynamically allocated and owned by the caller.

GetAtomLabelMonochrome

```
bool GetAtomLabelMonochrome() const
```

Returns whether or not the color of the atom label of the highlighted atom(s) is changed.

See also

- *OEHighlightByCogwheel.SetAtomLabelMonochrome* method

GetBallRadiusScale

```
double GetBallRadiusScale() const
```

Returns the multiplier that can be used to increase or decrease the ball radius of the 'cogwheel' of the highlighted atom(s).

See also

- *OEHighlightByCogwheel.SetBallRadiusScale* method

GetColor

```
const OESystem::OEColor& GetColor() const
```

Returns the color of the highlighting.

See also

- *OEHighlightByCogwheel.SetColor* method
- OEColor class

GetInnerContour

```
bool GetInnerContour() const
```

Returns whether the inner contour of the 'cogwheel' is drawn or committed.

See also

- *OEHighlightByCogwheel.SetInnerContour* method

GetLineWidthScale

```
double GetLineWidthScale() const
```

Returns the multiplier that can be used to increase or decrease the line width of the 'cogwheel'

See also

- *OEHighlightByCogwheel.SetLineWidthScale* method

GetStickWidthScale

```
double GetStickWidthScale() const
```

Returns the multiplier that can be used to increase or decrease the stick width of the 'cogwheel' of the highlighted bond(s).

See also

- *OEHighlightByCogwheel.SetStickWidthScale* method

SetAtomLabelMonochrome

```
void SetAtomLabelMonochrome(bool monochrome)
```

Sets whether or not to change the color of the atom label of the highlighted atom(s).

Example (*Figure: Example of using the SetAtomLabelMonochrome method*)

```
OEHighlightByCogwheel highlight = new OEHighlightByCogwheel(oechem.getOEDarkPurple());
highlight.SetAtomLabelMonochrome(false);
```

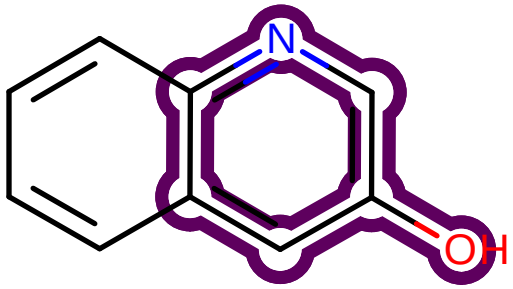


Fig. 86: **Example of using the SetAtomLabelMonochrome method**

See also

- *OEHighlightByCogwheel.GetAtomLabelMonochrome* method

SetBallRadiusScale

```
void SetBallRadiusScale(double ballRadiusScale)
```

Sets the multiplier that can be used to increase or decrease the ball radius of the 'cogwheel' of the highlighted atom(s).

Example (*Figure: Example of using the SetBallRadiusScale method*)

```
OEHighlightByCogwheel highlight = new OEHighlightByCogwheel(oechem.getOEDarkPurple());
highlight.SetBallRadiusScale(5.0);
```

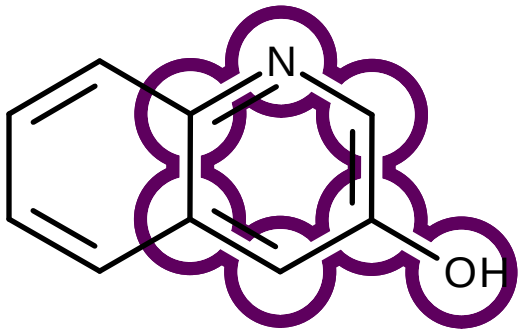


Fig. 87: **Example of using the SetBallRadiusScale method**

See also

- *OEHighlightByCogwheel.GetBallRadiusScale* method

SetColor

```
void SetColor(const OESystem::OEColor& color)
```

Sets the color of the highlighting.

Example (*Figure: Example of using the SetColor method*)

```
highlight.SetColor(oechem.getOEDarkPurple());
```

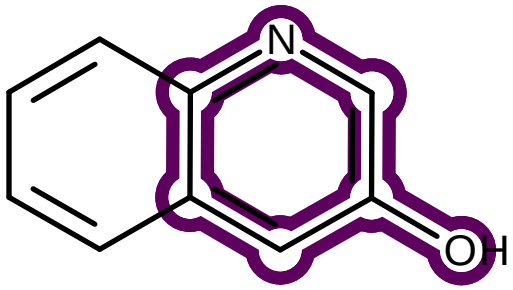


Fig. 88: **Example of using the SetColor method**

See also

- *OEHighlightByCogwheel.GetColor* method
- OEColor class

SetInnerContour

```
void SetInnerContour(bool innerContour)
```

Sets whether the inner contour of the 'cogwheel' is drawn or omitted.

Example (*Figure: Example of using the SetInnerContour method*)

```
OEHighlightByCogwheel highlight = new OEHighlightByCogwheel(oechem.getOEDarkPurple());
highlight.SetInnerContour(false);
```

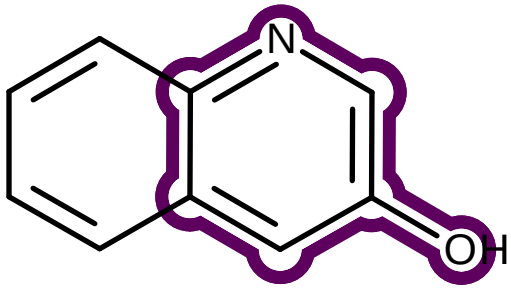


Fig. 89: **Example of using the SetInnerContour method**

See also

- *OEHighlightByCogwheel.GetInnerContour* method

SetLineWidthScale

```
void SetLineWidthScale(double lineWidthScale)
```

Sets the multiplier that can be used to increase or decrease the line width of the 'cogwheel'

Example (*Figure: Example of using the SetLineWidthScale method*)

```
OEHighlightByCogwheel highlight = new OEHighlightByCogwheel(oechem.getOEDarkPurple());
highlight.SetLineWidthScale(3.0);
```

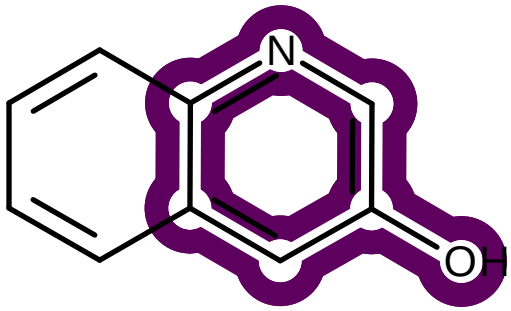


Fig. 90: **Example of using the SetLineWidthScale method**

See also

- *OEHighlightByCogwheel.GetLineWidthScale* method

SetStickWidthScale

```
void SetStickWidthScale(double stickWidthScale)
```

Sets the multiplier that can be used to increase or decrease the stick width of the 'cogwheel' of the highlighted bond(s).

Example (*Figure: Example of using the SetStickWidthScale method*)

```
OEHighlightByCogwheel highlight = new OEHighlightByCogwheel(oechem.getOEDarkPurple());
highlight.SetStickWidthScale(5.0);
```

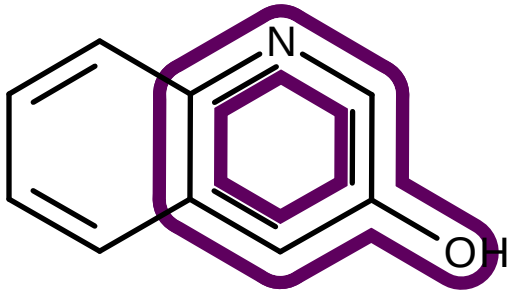


Fig. 91: **Example of using the SetStickWidthScale method**

See also

- *OEHighlightByCogwheel.GetStickWidthScale* method

4.1.30 OEHighlightByColor

```
class OEHighlightByColor : public OEHighlightBase
```

The *OEHighlightByColor* class allows the customization of the **color** highlighting style that is associated with the *OEHighlightStyle.Color* constant. See example in *Figure: Example of highlighting using the 'color' style*.

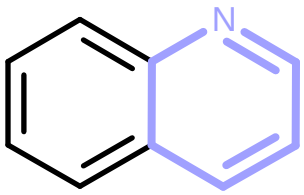


Fig. 92: **Example of highlighting using the 'color' style**

The *OEHighlightByColor* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
external highlight	*GetAtomExternalHigh*	*SetAtomExternalHigh*	boolean
color	*GetColor*	*SetColor*	OEColor
line width scale	*GetLineWidthScale*	*SetLineWidthScale*	floating point

See also

- *OEAddHighlighting* function
- *OEHighlightStyle* namespace
- *OEHighlightByBallAndStick* class
- *OEHighlightByCogwheel* class
- *OEHighlightByLasso* class
- *OEHighlightByStick* class
- *Highlighting* chapter

Constructors

```
OEHighlightByColor(const OESystem::OEColor& color,
                   double lineWidthScale = 1.5,
                   bool atomExternalHighlight = false)
```

Creates an *OEHighlightByColor* object with the specified parameters.

color
: The color used for highlighting. See also *OEHighlightByColor.SetColor* method.

lineWidthScale
: The multiplier that can be used to increase or decrease the line width of highlighted bond(s). See also *OEHighlightByColor.SetLineWidthScale* method.

atomExternalHighlight
: Sets whether atom highlights are extended to bonds not being highlighted. See also *OEHighlightByColor.SetAtomExternalHighlight* method.

```
OEHighlightByColor(const OEHighlightByColor& rhs)
```

Copy constructor.

CreateCopy

```
OEHighlightBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEHighlightByColor* object is dynamically allocated and owned by the caller.

GetAtomExternalHighlight

```
bool GetAtomExternalHighlight() const
```

Returns whether external atom highlight is applied.

See also

- *OEHighlightByColor.SetAtomExternalHighlight* method

GetColor

```
const OESystem::OEColor& GetColor() const
```

Returns the color of the highlighting.

See also

- *OEHighlightByColor.GetColor* method
- OEColor class

GetLineWidthScale

```
double GetLineWidthScale() const
```

Returns the multiplier that can be used to increase or decrease the line width of highlighted bond(s).

See also

- *OEHighlightByColor.SetLineWidthScale* method

SetAtomExternalHighlight

```
void SetAtomExternalHighlight(bool set)
```

Sets whether atom highlights are extended to bonds not being highlighted.

Example (*Figure: Example of using the SetAtomExternalHighlight method*)

```
OEHighlightByColor highlight = new OEHighlightByColor(oechem.getOELightOrange());
highlight.SetAtomExternalHighlight(true);
```

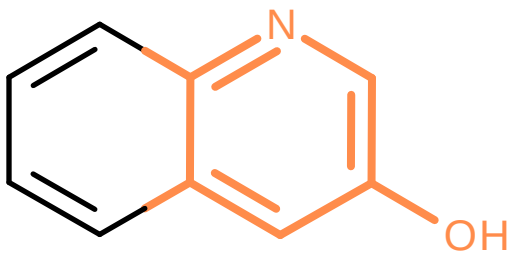


Fig. 93: **Example of using the SetAtomExternalHighlight method**

See also

- *OEHighlightByColor.GetAtomExternalHighlight* method

SetColor

```
void SetColor(const OESystem::OEColor& color)
```

Sets the color of the highlighting.

Example (*Figure: Example of using the SetColor method*)

```
highlight.SetColor(oechem.getOEDarkRed());
```

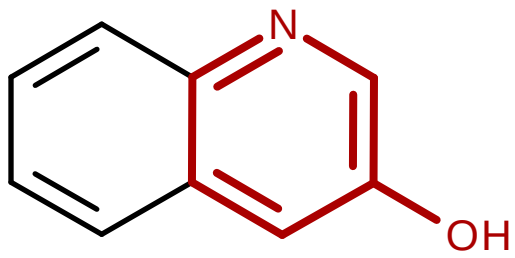


Fig. 94: **Example of using the SetColor method**

See also

- *OEHighlightByColor.GetColor* method
- OEColor class

SetLineWidthScale

```
void SetLineWidthScale(double lineWidthScale)
```

Sets the multiplier that can be used to increase or decrease the line width of highlighted bond(s).

Example (*Figure: Example of using the SetLineWidthScale method*)

```
OEHighlightByColor highlight = new OEHighlightByColor(oechem.getOEDarkRed());
highlight.SetLineWidthScale(3.0);
```

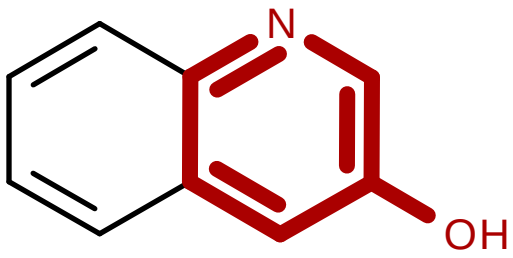


Fig. 95: **Example of using the SetLineWidthScale method**

See also

- *OEHighlightByColor.GetLineWidthScale* method

4.1.31 OEHighlightByLasso

```
class OEHighlightByLasso : public OEHighlightBase
```

The *OEHighlightByLasso* class allows the customization of the **lasso** highlighting style that is associated with the *OEHighlightStyle.Lasso* constant. See example in *Figure: Example of highlighting using the 'lasso' style*.

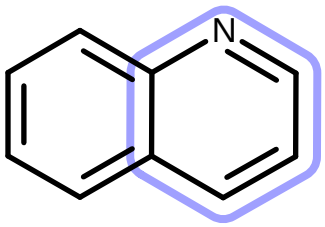


Fig. 96: **Example of highlighting using the 'lasso' style**

The *OEHighlightByLasso* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
atom label color	*GetAtomLabelMonochr*	*SetAtomLabelMonochr*	boolean
atom label box	*GetConsiderAtomLabe*	*SetConsiderAtomLabe*	boolean
color	*GetColor*	*SetColor*	OEColor
pen	*GetPen*	*SetPen*	*OEPen*
position	*GetPosition*	*SetPosition*	*OELayerPosition* namespace
lasso scale	*GetLassoScale*	*SetLassoScale*	floating point number
line width scale	*GetLineWidthScale*	*SetLineWidthScale*	floating point

See also

- *OEAddHighlighting* function
- *OEHighlightStyle* namespace
- *OEHighlightByBallAndStick* class
- *OEHighlightByColor* class
- *OEHighlightByCogwheel* class
- *OEHighlightByStick* class
- *Highlighting* chapter

Constructors

```
OEHighlightByLasso(const OESystem::OEColor& color,
                   double lassoScale = 3.0, bool monochrome = true,
                   bool considerAtomLabelBox = false,
                   unsigned int position = OELayerPosition::Below,
                   double lineWidthScale = 1.0)
```

Creates an *OEHighlightByLasso* object with the specified parameters.

color
: The color used for highlighting. See also *OEHighlightByLasso.SetColor* method.

lassoScale
: The multiplier that can be used to increase or decrease the distance of the "lasso" from the highlighted atom(s) and bond(s). See also *OEHighlightByLasso.SetLassoScale* method.

monochrome
: Defines whether or not to change the color of the atom label of the highlighted atom(s). See also *OEHighlightByLasso.SetAtomLabelMonochrome* method.

considerAtomLabelBox
: Determines whether or not to consider the atom label bounding boxes when generating the 'lasso' highlights. See also *OEHighlightByLasso.SetConsiderAtomLabelBoundingBox* method.

position
: Defines whether the 'lasso' highlight is going to be rendered below or above the molecule. See also *OEHighlightByLasso.SetPosition* method.

lineWidthScale
: The multiplier that can be used to increase or decrease the line width of lasso. See also *OEHighlightByLasso.SetLineWidthScale* method.

```
OEHighlightByLasso(const OEHighlightByLasso& rhs)
```

Copy constructor.

CreateCopy

```
OEHighlightBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEHighlightByLasso* object is dynamically allocated and owned by the caller.

GetAtomLabelMonochrome

```
bool GetAtomLabelMonochrome() const
```

Returns whether or not the color of the atom label of the highlighted atom(s) is changed.

See also

- *OEHighlightByLasso.SetAtomLabelMonochrome* method

GetConsiderAtomLabelBoundingBox

```
bool GetConsiderAtomLabelBoundingBox() const
```

Returns whether or not to consider the atom label bounding boxes when generating the 'lasso' highlights.

See also

- *OEHighlightByLasso.SetConsiderAtomLabelBoundingBox* method

GetLassoScale

```
double GetLassoScale() const
```

Returns the multiplier that can be used to increase or decrease the distance of the line of the 'lasso' from the highlighted atom(s) and bonds(s).

See also

- *OEHighlightByLasso.SetLassoScale* method

GetLineWidthScale

```
double GetLineWidthScale() const
```

Returns the multiplier that can be used to increase or decrease the line width of the lasso.

See also

- *OEHighlightByLasso.SetLineWidthScale* method

GetColor

```
const OESystem::OEColor& GetColor() const
```

Returns the foreground color of the highlighting.

See also

- *OEHighlightByLasso.GetColor* method
- OEColor class

GetPen

```
const OEPen& GetPen() const
```

Returns the pen of the highlighting.

See also

- *OEHighlightByLasso.SetPen* method
- *OEPen* class

GetPosition

```
unsigned int GetPosition() const
```

Returns whether the 'lasso' highlight is going to be render above or below the molecule. The return value is taken from the *OELayerPosition* namespace.

See also

- *OEHighlightByLasso.SetPosition* method

- *OELayerPosition* namespace

SetAtomLabelMonochrome

```
void SetAtomLabelMonochrome(bool monochrome)
```

Sets whether or not to change the color of the atom label of the highlighted atom(s).

Example (*Figure: Example of using the SetAtomLabelMonochrome method*)

```
OEHighlightByLasso highlight = new OEHighlightByLasso(oechem.getOEDarkGreen());
highlight.SetAtomLabelMonochrome(false);
```

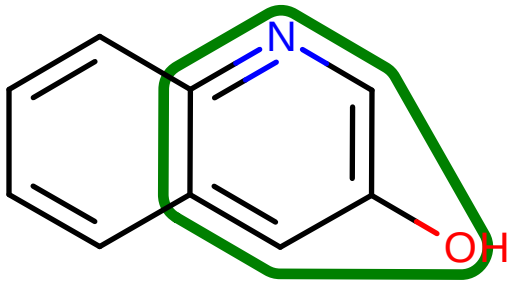


Fig. 97: **Example of using the SetAtomLabelMonochrome method**

See also

- *OEHighlightByLasso.GetAtomLabelMonochrome* method

SetConsiderAtomLabelBoundingBox

```
void SetConsiderAtomLabelBoundingBox(bool consider)
```

Sets whether or not to consider the atom label bounding boxes when generating the 'lasso' highlights. By considering the bounding boxes, the number cases when the lines of the highlight clash with the atom label can be reduced.

Example (*Figure: Example of using the SetConsiderAtomLabelBoundingBox method*)

```
OEHighlightByLasso highlight = new OEHighlightByLasso(oechem.getOEDarkGreen());
highlight.SetConsiderAtomLabelBoundingBox(true);
```

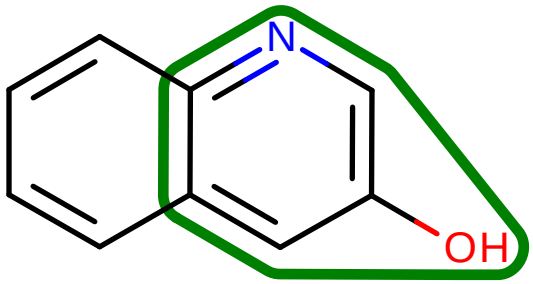


Fig. 98: **Example of using the SetConsiderAtomLabelBoundingBox method**

See also

- *OEHighlightByLasso.SetConsiderAtomLabelBoundingBox* method

SetLassoScale

```
void SetLassoScale(double lassoScale)
```

Sets the multiplier that can be used to increase or decrease the distance of the line of the 'lasso' from the highlighted atom(s) and bond(s).

Example (*Figure: Example of using the SetLassoScale method*)

```
OEHighlightByLasso highlight = new OEHighlightByLasso(oechem.getOEDarkGreen());
highlight.SetLassoScale(6.0);
```

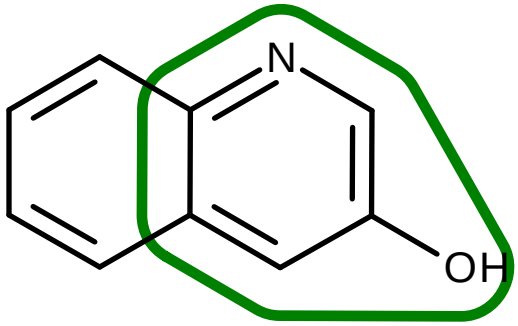


Fig. 99: **Example of using the SetLassoScale method**

See also

- *OEHighlightByLasso.GetLassoScale* method

SetLineWidthScale

```
void SetLineWidthScale(double lineWidthScale)
```

Sets the multiplier that can be used to increase or decrease the line width of the lasso.

Example (*Figure: Example of using the SetLineWidthScale method*)

```
OEHighlightByLasso highlight = new OEHighlightByLasso(oechem.getOEDarkGreen());
highlight.SetLineWidthScale(3.0);
```

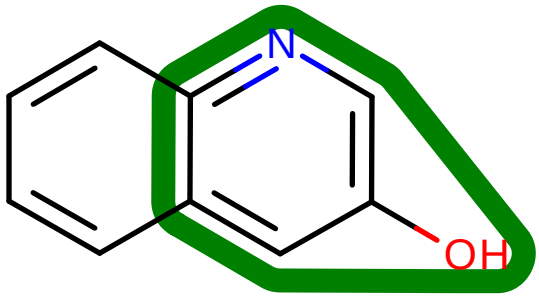


Fig. 100: **Example of using the SetLineWidthScale method**

See also

- *OEHighlightByLasso.GetLineWidthScale* method

SetColor

```
void SetColor(const OESystem::OEColor& color)
```

Sets the foreground color of the highlighting.

Example (*Figure: Example of using the SetColor method*)

```
highlight.SetColor(oechem.getOEDarkGreen());
```

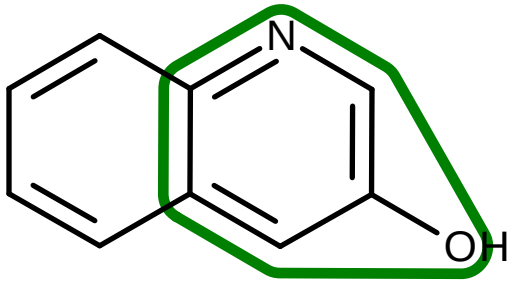


Fig. 101: **Example of using the SetColor method**

See also

- *OEHighlightByLasso.GetColor* method
- OEColor class

SetPen

```
void SetPen(const OEPen& pen)
```

Sets the pen of the highlighting.

Example (*Figure: Example of using the SetPen method*)

```
OEPen pen = new OEPen(oechem.getOEGreenTint(), oechem.getOEDarkGreen(), OEFill.On, 3.0,␣
↪OEStipple.ShortDash);
highlight.SetPen(pen);
```

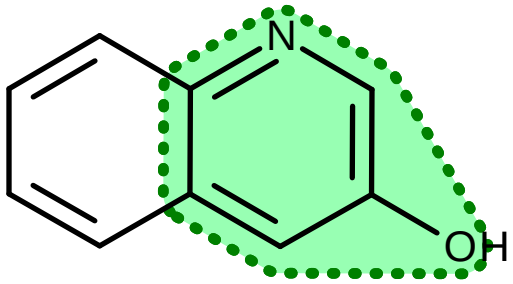


Fig. 102: **Example of using the SetPen method**

See also

- *OEHighlightByLasso.GetPen* method
- *OEPen* class

SetPosition

```
void SetPosition(unsigned int position)
```

Sets whether the 'lasso' highlight is going to be render above or below the molecule.

position
This value has to be from the *OELayerPosition* namespace.

See also

- *OEHighlightByLasso.GetPosition* method

4.1.32 OEHighlightByStick

```
class OEHighlightByStick : public OEHighlightBase
```

The *OEHighlightByStick* class allows the customization of the **stick** highlighting style that is associated with the *OEHighlightStyle.Stick* constant. See example in *Figure: Example of highlighting using the 'stick' style*.

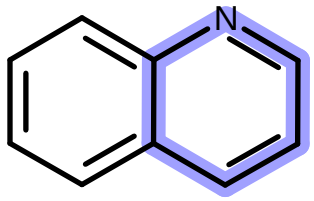


Fig. 103: **Example of highlighting using the 'stick' style**

The *OEHighlightByStick* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
external highlight	*GetAtomExternalHigh*	*SetAtomExternalHigh*	floating point
atom label	*GetAtomLabelMonochr*	*SetAtomLabelMonochr*	boolean
color	*GetColor*	*SetColor*	OEColor
stick width scale	*GetStickWidthScale*	*SetStickWidthScale*	floating point number

See also

- *OEAddHighlighting* function
- *OEHighlightStyle* namespace
- *OEHighlightByBallAndStick* class
- *OEHighlightByColor* class
- *OEHighlightByCogwheel* class
- *OEHighlightByLasso* class
- *Highlighting* chapter

Constructors

```
OEHighlightByStick(const OESystem::OEColor& color,
                   double stickWidthScale = 3.0,
                   bool monochrome = true,
                   double atomExternalHighlightRatio = 0.0)
```

Creates an *OEHighlightByStick* object with the specified parameters.

color
The color used for highlighting. See also *OEHighlightByStick.SetColor* method.

stickWidthScale
The multiplier that can be used to increase or decrease the stick width of the highlighted bond(s). See also *OEHighlightByStick.SetStickWidthScale* method.

monochrome
Defines whether or not to change the color of the atom label of the highlighted atom(s). See also *OEHighlightByStick.SetAtomLabelMonochrome* method.

atomExternalHighlightRatio
A ratio that can be used to extend atom highlights to bonds not being highlighted. See also *OEHighlightByStick.SetAtomExternalHighlightRatio* method.

```
OEHighlightByStick(const OEHighlightByStick& rhs)
```

Copy constructor.

CreateCopy

```
OEHighlightBase* CreateCopy() const
```

Deep copy constructor that returns a copy of the object. The memory for the returned *OEHighlightByStick* object is dynamically allocated and owned by the caller.

GetAtomExternalHighlightRatio

```
double GetAtomExternalHighlightRatio() const
```

Returns the ratio of the external highlight of the atoms.

See also

- *OEHighlightByStick.SetAtomExternalHighlightRatio* method

GetAtomLabelMonochrome

```
bool GetAtomLabelMonochrome() const
```

Returns whether or not the color of the atom label of the highlighted atom(s) is changed.

See also

- *OEHighlightByStick.SetAtomLabelMonochrome* method

GetColor

```
const OESystem::OEColor& GetColor() const
```

Returns the color of the highlighting.

See also

- *OEHighlightByStick.SetColor* method
- OEColor class

GetStickWidthScale

```
double GetStickWidthScale() const
```

Returns the multiplier that can be used to increase or decrease the stick width of the highlighted bond(s).

See also

- *OEHighlightByStick.SetStickWidthScale* method

SetAtomExternalHighlightRatio

```
void SetAtomExternalHighlightRatio(double ratio)
```

Sets the ratio of the external highlight of the atoms *i.e.* atom highlights are extended to bonds not being highlighted.

ratio
: This value has to be in the range of `[0.0, 0.5]`.

Example (*Figure: Example of using the SetAtomExternalHighlightRatio method*)

```
OEHighlightByStick highlight = new OEHighlightByStick(oechem.getOELightOrange());
highlight.SetAtomExternalHighlightRatio(0.33);
```

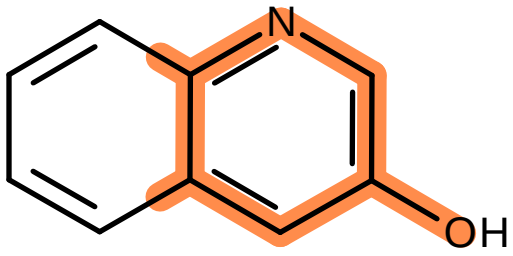


Fig. 104: **Example of using the SetAtomExternalHighlightRatio method**

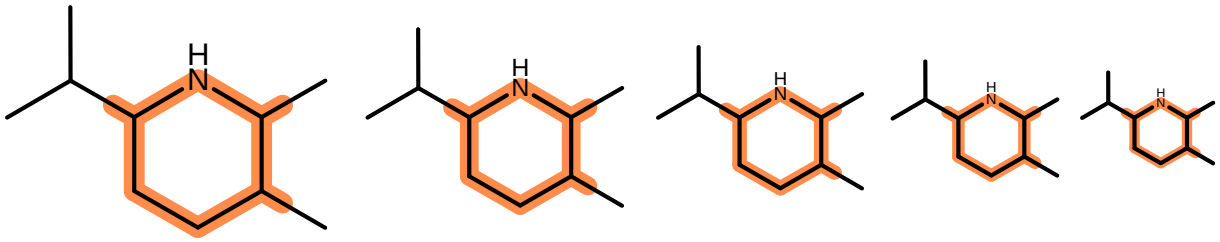


Fig. 105: **Example of scaling the 'Stick' style highlighting along with the molecule**

See also

- *OEHighlightByStick.GetColor* method

SetAtomLabelMonochrome

```
void SetAtomLabelMonochrome(bool monochrome)
```

Sets whether or not to change the color of the atom label of the highlighted atom(s).

Example (*Figure: Example of using the SetAtomLabelMonochrome method*)

```
OEHighlightByStick highlight = new OEHighlightByStick(oechem.getOELightOrange());
highlight.SetAtomLabelMonochrome(false);
```

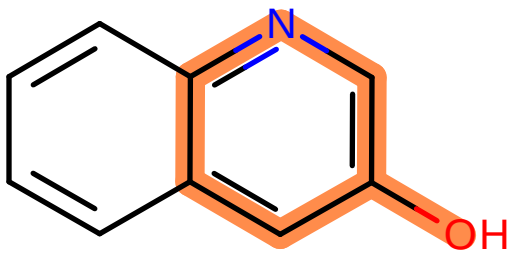


Fig. 106: **Example of using the SetAtomLabelMonochrome method**

See also

- *OEHighlightByStick.GetAtomLabelMonochrome* method

SetColor

```
void SetColor(const OESystem::OEColor& color)
```

Sets the color of the highlighting.

Example (*Figure: Example of using the SetColor method*)

```
highlight.SetColor(oechem.getOELightOrange());
```

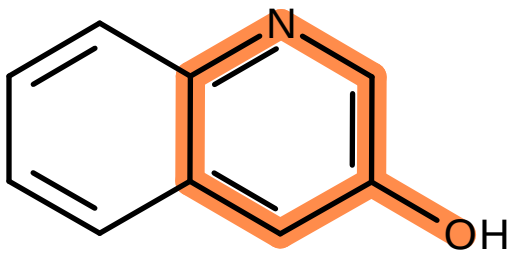


Fig. 107: **Example of using the SetColor method**

See also

- *OEHighlightByStick.GetColor* method
- OEColor class

SetStickWidthScale

```
void SetStickWidthScale(double stickWidthScale)
```

Sets the multiplier that can be used to increase or decrease the stick width of the highlighted bond(s).

Example (*Figure: Example of using the SetStickWidthScale method*)

```
OEHighlightByStick highlight = new OEHighlightByStick(oechem.getOELightOrange());
highlight.SetStickWidthScale(5.0);
```

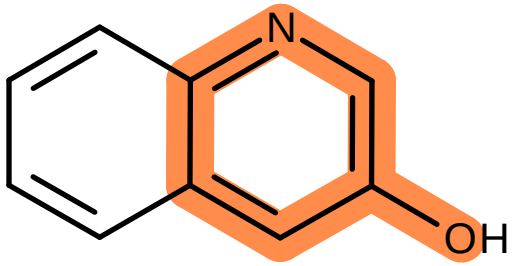


Fig. 108: **Example of using the SetStickWidthScale method**

See also

- *OEHighlightByStick.GetStickWidthScale* method

4.1.33 OEHighlightLabel

```
class OEHighlightLabel
```

The *OEHighlightLabel* class allows the customization the style of the label that can be position on molecule depiction by calling the *OEAddLabel* function.

The *OEHighlightLabel* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
text	*GetText*	*OEHighlightLabel constructors*	string
bounding box pen	*GetBoundingBoxPen*	*SetBoundingBoxPen*	*OEPen*
label font	*GetFont*	*SetFont*	*OEFont*
label font scale	*GetFontScale*	*SetFontScale*	positive floating point number in the range of `[0.25, 1.5]` or `0.0`

Constructors

```
OEHighlightLabel(const std::string& text)
```

Creates an *OEHighlightLabel* object with default style.

text
The text that is going to be positioned on the molecule depiction.

Example:

```
OEHighlightLabel label = new OEHighlightLabel("match");
```

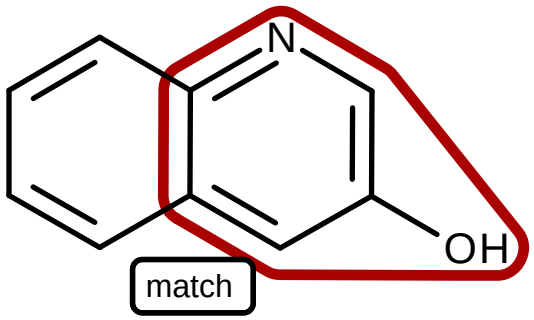


Fig. 109: **Example of adding label with default style**

```
OEHighlightLabel(const std::string& text, const OEFont& font)
```

Creates an *OEHighlightLabel* object with user-defined font.

text
The text that is going to be positioned on the molecule depiction.

font
The *OEFont* object that encapsulates properties that determine the display style of text.

See also

- *OEFont* class

Example:

```
OEFont font = new OEFont(OEFontFamily.Courier, OEFontStyle.Bold, 12,
                         OEAlignment.Center, oechem.getOEDarkRed());
OEHighlightLabel label = new OEHighlightLabel("match", font);
```

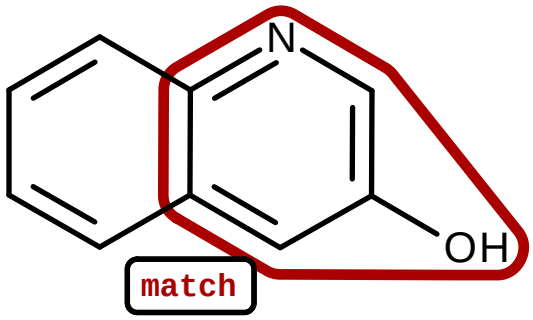


Fig. 110: **Example of adding label with user-defined font**

```
OEHighlightLabel(const std::string& text,
                 const OESystem::OEColor& color)
```

Creates an *OEHighlightLabel* object with user-defined border color.

text
The text that is going to be positioned on the molecule depiction.

color
The color of the border around the label.

Example:

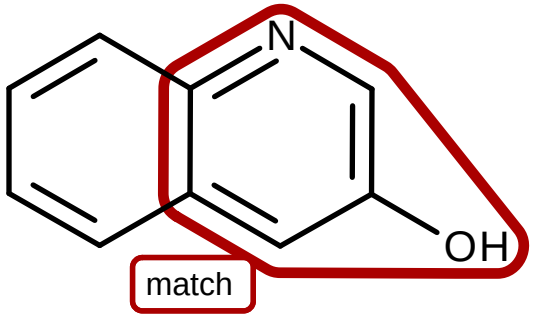


Fig. 111: **Example of adding label with user-defined border color**

See also

- OEColor class

GetBoundingBoxPen

```
const OEPen& GetBoundingBoxPen() const
```

Returns the pen that is used to draw a border around the label.

See also

- *OEPen* class
- *SetBoundingBoxPen* method

GetFont

```
const OEFont& GetFont() const
```

Returns the font of the label.

See also

- *OEFont* class
- *OEHighlightLabel.SetFont* method

GetFontScale

```
double GetFontScale() const
```

Returns the multiplier that can be used to increase or decrease the font size of the label relative to the font size used to render labels of the of the depicted molecule. The default value is `0.75`.

See also

- *OEFont* class
- *OEHighlightLabel.SetFontScale* method

GetText

```
std::string GetText() const
```

Returns the text of the label.

SetBoundingBoxPen

```
void SetBoundingBoxPen(const OEPen& pen)
```

Sets the pen that is used to draw a border around the label.

Example:

```
OEHighlightLabel label = new OEHighlightLabel("match");
OEPen pen = new OEPen(new OEColor(255, 220, 220), oechem.getOEDarkRed(), OEFill.On, 16.0,
↪ OEStipple.ShortDash);
label.SetBoundingBoxPen(pen);
```

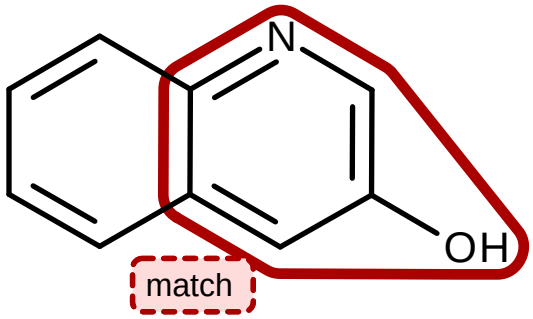


Fig. 112: **Example of adding label with user-defined border**

See also

- *OEPen* class

SetFont

```
void SetFont(const OEFont& font)
```

Set the font of the label.

Example:

```
OEHighlightLabel label = new OEHighlightLabel("match");
OEFont font = new OEFont(OEFontFamily.Default, OEFontStyle.Default, 12,
                         OEAlignment.Center, oechem.getOEDarkRed());
label.SetFont(font);
```

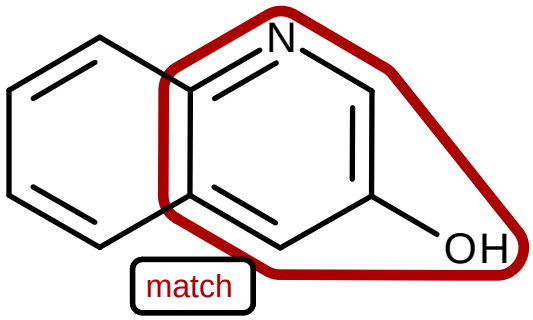


Fig. 113: **Example of adding label with user-defined font**

Note

The size of fonts of the label depends on:

- the size of the font used to render atom labels of the the depicted molecule
- the multiplier set by the *OEHighlightLabel.SetFontScale* method

See also

- *OEFont* class

SetFontScale

```
void SetFontScale(double scale)
```

Sets the multiplier that can be used to increase or decrease the font size of the label relative to the font size used to render atom labels of the of the depicted molecule.

scale

This value has to be either `0.0` or in a range of `[0.25, 1.50]`. The default value is `0.75`.

In case of `0.0`, the label is not scaled with the molecule but rather fixed font size is used regardless of the size of the depicted molecule.

Example:

```
OEHighlightLabel label = new OEHighlightLabel("match");
label.SetFontScale(1.2);
```

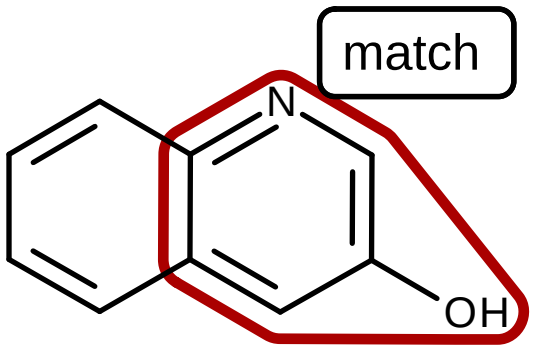


Fig. 114: **Example of adding label with user-defined font scaling**

4.1.34 OEHighlightOverlayBase

```
class OEHighlightOverlayBase
```

The *OEHighlightOverlayBase* is an abstract class used for highlighting a set of atoms and/or bonds.

The following classes derive from this class:

- *OEHighlightOverlayByBallAndStick*

See also

- *OEAddHighlightOverlay* function
- *Highlighting Overlapped Patterns* section

Constructors

```
OEHighlightOverlayBase()
OEHighlightOverlayBase(OEHighlightOverlayBaseImpl* impl)
```

Default constructors.

GetColors

```
OESystem::OEIterBase<OESystem::OEColor>* GetColors() const
```

Returns an iterator over the colors used for highlighting.

SetColors

```
void SetColors(OESystem::OEIter<OESystem::OEColor>& colors)
void SetColors(OESystem::OEIterBase<OESystem::OEColor>* colors)
```

Sets the colors used for highlighting.

4.1.35 OEHighlightOverlayByBallAndStick

```
class  OEHighlightOverlayByBallAndStick : public OEHighlightOverlayBase
```

The *OEHighlightOverlayByBallAndStick* class allows the customization of the **ball and stick** highlighting style that is associated with the *OEHighlightOverlayStyle.BallAndStick* constant. This style enables the highlighting of overlapping patterns. See example in *Figure: Example of overlay highlighting using the 'ball and stick' style*.

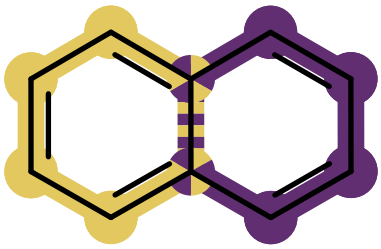


Fig. 115: **Example of overlay highlighting using the 'ball and stick' style**

See also

- *OEAddHighlightOverlay* function.
- *OEHighlightOverlayStyle* namespace
- *Highlighting Overlapped Patterns* section

Note

It is recommended to select colors with high contrast when highlighting overlapped patterns with the *OEHighlightOverlayByBallAndStick* class.

See also

- OEGetContrastColors function

The following methods are publicly inherited from *OEHighlightOverlayBase*:

GetColors	*SetColors*

Constructors

```
OEHighlightOverlayByBallAndStick(OESystem::OEIter<OESystem::OEColor>& colors,
                                 double stickWidthScale = 3.0,
                                 double ballRadiusScale = 3.0,
                                 bool monochrome = true)
OEHighlightOverlayByBallAndStick(OESystem::OEIterBase<OESystem::OEColor>* colors,
```

(continues on next page)

(continued from previous page)

```
                                    double stickWidthScale = 3.0,
                                    double ballRadiusScale = 3.0,
                                    bool monochrome = true)
```

Creates an *OEHighlightOverlayByBallAndStick* object with the specified parameters

colors
: The colors used for highlighting. See also *OEHighlightOverlayBase.SetColors* method.

stickWidthScale
: The multiplier that can be used to increase or decrease the stick width of the highlighted bond(s). See also *OEHighlightOverlayByBallAndStick.SetStickWidthScale* method.

ballRadiusScale
: The multiplier that can be used to increase or decrease the ball radius of the highlighted atoms(s). See also *OEHighlightOverlayByBallAndStick.SetBallRadiusScale* method.

monochrome
: Defines whether or not to change the color of the atom label of the highlighted atom(s). See also *OEHighlightOverlayByBallAndStick.SetAtomLabelMonochrome* method.

```
OEHighlightOverlayByBallAndStick(const OEHighlightOverlayByBallAndStick& rhs)
```

Copy constructor.

GetAtomLabelMonochrome

```
bool GetAtomLabelMonochrome() const
```

Returns whether or not the color of the atom label of the highlighted atom(s) is changed.

See also

- *OEHighlightOverlayByBallAndStick.SetAtomLabelMonochrome* method

GetBallRadiusScale

```
double GetBallRadiusScale() const
```

Returns the multiplier that can be used to increase or decrease the ball radius of the highlighted atoms(s).

See also

- *OEHighlightOverlayByBallAndStick.SetBallRadiusScale* method

GetStickWidthScale

```
double GetStickWidthScale() const
```

Returns the multiplier that can be used to increase or decrease the stick width of the highlighted bond(s).

See also

- *OEHighlightOverlayByBallAndStick.SetStickWidthScale* method

SetAtomLabelMonochrome

```
void SetAtomLabelMonochrome(bool monochrome)
```

Sets whether or not to change the color of the atom label of the highlighted atom(s).

Example (*Figure: Example of using the SetAtomLabelMonochrome method*)

```
OEHighlightOverlayByBallAndStick highlight =
  new OEHighlightOverlayByBallAndStick(oechem.OEGetVividColors());
highlight.SetAtomLabelMonochrome(false);
depictWithHighlight(smiles, ss, highlight, "OEHighlightOverlayByBallAndStick_
↪SetAtomLabelMonochrome");
```

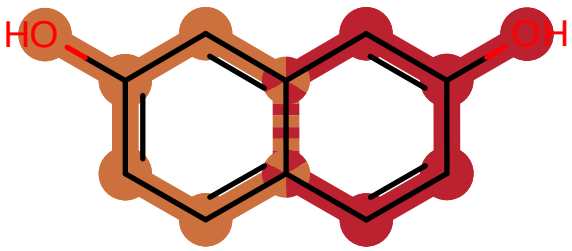


Fig. 116: **Example of using the SetAtomLabelMonochrome method**

See also

- *OEHighlightOverlayByBallAndStick.GetAtomLabelMonochrome* method

SetBallRadiusScale

```
void SetBallRadiusScale(double ballRadiusScale)
```

Sets the multiplier that can be used to increase or decrease the ball radius of the highlighted atoms(s).

Example (*Figure: Example of using the SetBallRadiusScale method*)

```
OEHighlightOverlayByBallAndStick highlight =
  new OEHighlightOverlayByBallAndStick(oechem.OEGetVividColors());
highlight.SetBallRadiusScale(4.0);
```

(continues on next page)

(continued from previous page)

```
depictWithHighlight(smiles, ss, highlight, "OEHighlightOverlayByBallAndStick_
↪SetBallRadiusScale");
```

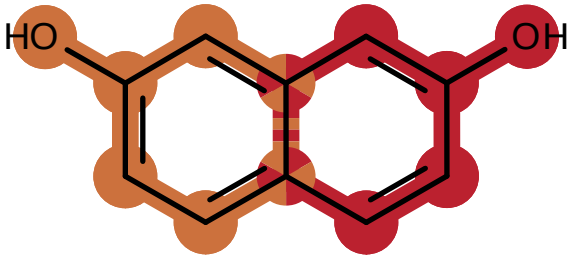


Fig. 117: **Example of using the SetBallRadiusScale method**

See also

- *OEHighlightOverlayByBallAndStick.GetBallRadiusScale* method

SetStickWidthScale

```
void SetStickWidthScale(double stickWidthScale)
```

Sets the multiplier that can be used to increase or decrease the stick width of the highlighted bond(s).

Example (*Figure: Example of using the SetStickWidthScale method*)

```
OEHighlightOverlayByBallAndStick highlight =
  new OEHighlightOverlayByBallAndStick(oechem.OEGetVividColors());
highlight.SetStickWidthScale(5.0);
depictWithHighlight(smiles, ss, highlight, "OEHighlightOverlayByBallAndStick_
↪SetStickWidthScale");
```

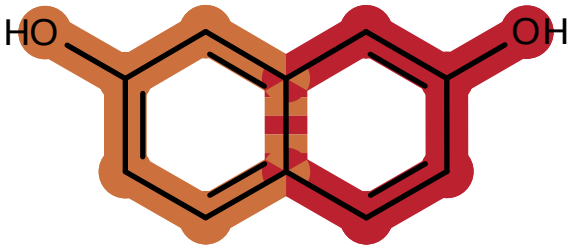


Fig. 118: **Example of using the SetStickWidthScale method**

See also

- *OEHighlightOverlayByBallAndStick.GetStickWidthScale* method

4.1.36 OEImage

```
class OEImage : public OEImageBase
```

OEImage class is implemented as a **display list** i.e. a container of drawing commands. For example, when the *OEImage.DrawLine* method is called, rather than immediately drawing a line specified by the parameters, a line drawing operation is created that copies all parameters for later execution.

When the *OEImage.Render* method is invoked, the drawing commands stored in the *OEImage* object are executed in the same order in which they were issued.

The following methods are publicly inherited from *OEImageBase*:

Clear	*DrawPie*	*DrawTriangle*
DrawArc	*DrawPoint*	*GetGlobalOffset*
DrawCircle	*DrawPolygon*	*GetHeight*
DrawCubicBezier	*DrawQuadraticBezier*	*GetMinFontSize*
DrawLine	*DrawRectangle*	*GetWidth*
DrawPath	*DrawText*	*GetSVGClass*
GetSVGGroup	*GetSVGGroups*	*NewSVGClass*
NewSVGGroup	*PopGroup*	*PushGroup*
SetMinFontSize		

Constructors

```
OEImage(double width, double height,
        const OESystem::OEColor& bgColor = OESystem::OEWhite)
```

Default constructor that creates an *OEImage* with the specified width and height.

width, height
The dimensions of the image, both have to be positive (non-zero) numbers.

bgColor
The color that is used to clear the image upon construction.

The default background color is `OEWhite`.
A `.png` image with transparent background can be generated by passing the

`OETransparentColor`
as the background color.

```
OEImage(const OEImage& rhs)
```

Copy constructor.

```
OEImage(const OEImage& src, double scale)
```

Copy constructor with scaling.

scale
The scaling factor is used to create a new image from the given one. The scaling factor has to be a positive (non-zero) number.

The following code snippet shows how to half and double the size of an image.

```
OEImage image = new OEImage(100.0, 100.0);

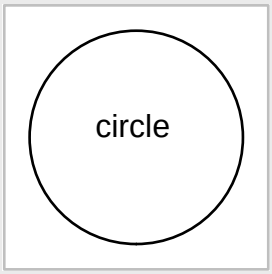
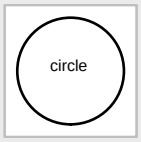
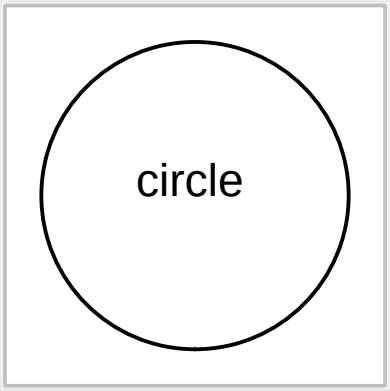
image.DrawCircle(oedepict.OEGetCenter(image), 40.0, oedepict.getOEBlackPen());
image.DrawText(oedepict.OEGetCenter(image), "circle", oedepict.getOEDefaultFont());
oedepict.OEDrawBorder(image, oedepict.getOELightGreyPen());

oedepict.OEWriteImage("image.png", image);

OEImage halfimage = new OEImage(image, 0.5);
oedepict.OEWriteImage("halfimage.png", halfimage);

OEImage doubleimage = new OEImage(image, 2.0);
oedepict.OEWriteImage("doubleimage.png", doubleimage);
```

Table 7: **Examples of image scaling**

original image	scale = 0.5	scale = 2.0
		

Clear

```
void Clear(const OESystem::OEColor& color)
```

Appends a clear command to the display list of the *OEImage* object.

See also

- *OEImageBase.Clear* method
- OEColor class

DrawArc

```
void DrawArc(const OE2DPoint& center, double bgnAngle,
             double endAngle, double radius, const OEPen& pen)
```

Appends an arc drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing an arc*.

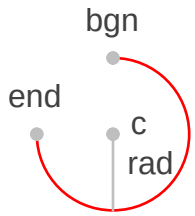


Fig. 119: **Example of drawing an arc**

See also

- *OEImageBase.DrawArc* method
- *OE2DPoint* class
- *OEPen* class

DrawCircle

```
void DrawCircle(const OE2DPoint& center, double radius,
                const OEPen& pen)
```

Appends a circle drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing a circle*.

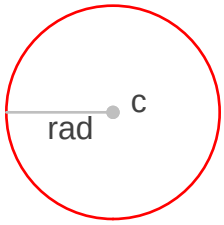


Fig. 120: **Example of drawing a circle**

See also

- *OEImageBase.DrawCircle* method
- *OE2DPoint* class
- *OEPen* class

DrawCubicBezier

```
void DrawCubicBezier(const OE2DPoint& bgn, const OE2DPoint& c1,
                     const OE2DPoint& c2, const OE2DPoint& end,
                     const OEPen& pen)
```

Appends a cubic Bézier curve drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing a cubic Bezier curve*.

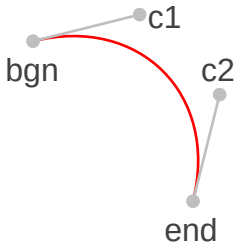


Fig. 121: **Example of drawing a cubic Bezier curve**

See also

- *OEImageBase.DrawCubicBezier* method
- *OE2DPoint* class
- *OEPen* classes

DrawLine

```
void DrawLine(const OE2DPoint& bgn, const OE2DPoint& end,
              const OEPen& pen)
```

Appends a line drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing a line*.

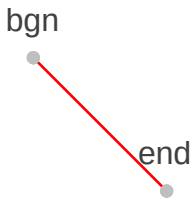


Fig. 122: **Example of drawing a line**

See also

- *OEImageBase.DrawLine* method
- *OE2DPoint* class
- *OEPen* class

DrawPath

```
void DrawPath(const OE2DPath& path, const OEPen& pen)
```

Appends a path drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing a path*.

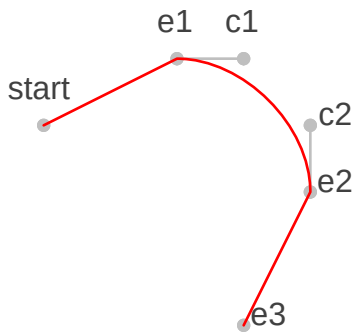


Fig. 123: **Example of drawing a path**

See also

- *OEImageBase.DrawPath* method
- *OE2DPath* class
- *OEPen* class

DrawPie

```
void DrawPie(const OE2DPoint& center, double bgnAngle,
             double endAngle, double radius, const OEPen& pen)
```

Appends a pie drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing a pie*.

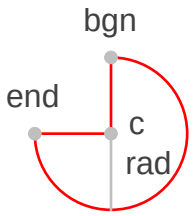


Fig. 124: **Example of drawing a pie**

See also

- *OEImageBase.DrawPie* method
- *OE2DPoint* class
- *OEPen* class

DrawPoint

```
void DrawPoint(const OE2DPoint& p, const OESystem::OEColor& color)
```

Appends a point drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing a point*.

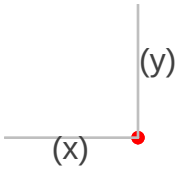


Fig. 125: **Example of drawing a point**

See also

- *OEImageBase.DrawPoint* method
- *OE2DPoint* class
- OEColor class

DrawPolygon

```
void DrawPolygon(const std::vector<OE2DPoint>& points,
                 const OEPen& pen)
```

Appends a polygon drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing a polygon*.

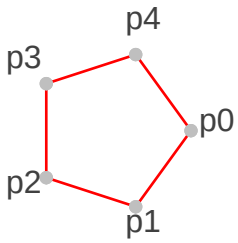


Fig. 126: **Example of drawing a polygon**

See also

- *OEImageBase.DrawPolygon* method
- *OE2DPoint* class

- *OEPen* class

DrawQuadraticBezier

```
void DrawQuadraticBezier(const OE2DPoint& bgn, const OE2DPoint& c,
                         const OE2DPoint& end, const OEPen& pen)
```

Appends a quadratic Bézier curve drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing a quadratic Bezier curve*.

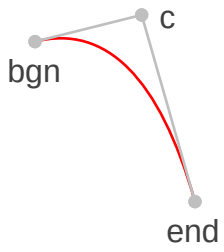


Fig. 127: **Example of drawing a quadratic Bezier curve**

See also

- *OEImageBase.DrawQuadraticBezier* method
- *OE2DPoint* class
- *OEPen* classes

DrawRectangle

```
void DrawRectangle(const OE2DPoint& tl, const OE2DPoint& br,
                   const OEPen& pen)
```

Appends a polygon drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing a rectangle*.

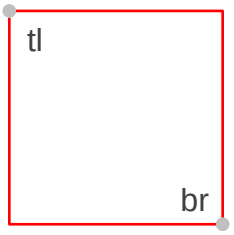


Fig. 128: **Example of drawing a rectangle**

See also

- *OEImageBase.DrawRectangle* method
- *OE2DPoint* class
- *OEPen* class

DrawText

```
void DrawText(const OE2DPoint& c, const std::string& text,
              const OEFont& font, double maxwidth = 0.0)
```

Appends a text drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing a text*.



Fig. 129: **Example of drawing a text**

See also

- *OEImageBase.DrawText* method
- *OE2DPoint* class
- *OEFont* class

DrawTriangle

```
void DrawTriangle(const OE2DPoint& a, const OE2DPoint& b,
                  const OE2DPoint& c, const OEPen& pen)
```

Appends a triangle drawing command to the display list of the *OEImage* object. See example in *Figure: Example of drawing a triangle*.

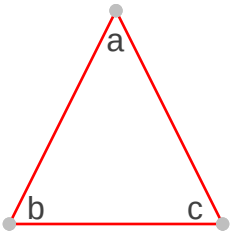


Fig. 130: **Example of drawing a triangle**

See also

- *OEImageBase.DrawTriangle* method
- *OE2DPoint* class
- *OEPen* class

GetHeight

```
double GetHeight() const
```

Returns the height of the *OEImage* object.

GetWidth

```
double GetWidth() const
```

Returns the width of the *OEImage* object.

IsEmpty

```
bool IsEmpty() const
```

Returns whether or not the display list of the *OEImage* object is empty.

Render

```
bool Render(OEImageBase& image) const
```

Renders the image stored in the *OEImage* object by executing the stored drawing commands in the order in which they were issued.

4.1.37 OEImageBase

```
class OEImageBase
```

The *OEImageBase* is an abstract class that provides methods for drawing basic geometric shapes such as lines, circles, rectangles etc.

The following classes derive from this class:

- *OEImage*
- *OEImageFrame*

Constructors

```
OEImageBase(double width = 200.0, double height = 200.0)
```

Default constructor that creates an image with the specified width and height.

width, height
The dimensions of the image, both have to be positive (non-zero) numbers.

Clear

```
void Clear(const OESystem::OEColor& color = OESystem::OEWhite)
```

Clears the image with the given color.

See also

- OEColor class

DrawArc

```
void DrawArc(const OE2DPoint& center, double bgnAngle,
             double endAngle, double radius, const OEPen& pen)
```

Draws the outline of an arc. See example in *Figure: Example of arc*.

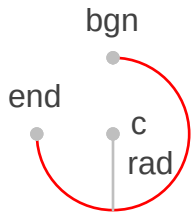


Fig. 131: **Example of arc**

center
The center of the arc.

bgnAngle, endAngle
The two endpoints of the arc (in degrees). Both angles are in degrees and their values have to be in a range of $[0.0°, 360.0°]$. Angles are interpreted such that $0.0°$ and $360.0°$ degrees are at the *12 o'clock* position, $90.0°$ degrees corresponds to *3 o'clock*, etc.

radius
The radius of the arc.

pen
The graphical properties of the curve (such as color, line width, etc). See examples in *Figure: Example of drawing arcs with various pens*.

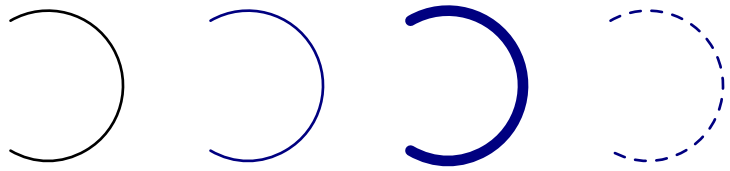


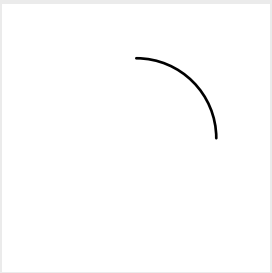
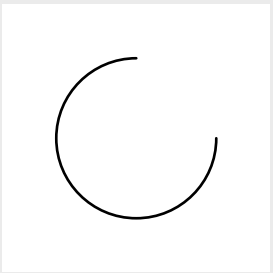
Fig. 132: **Example of drawing arcs with various pens**

Note

OEImageBase.DrawArc supports all the *OEPen* properties (i.e. color, line width, line stipple) except the fill property. See also *OEImageBase.DrawPie* method.

Arcs are always drawn in a clockwise direction. This means that bgn = 0.0°, end = 90.0° angle pair is not the same as bgn = 90.0°, end = 0.0° angle pair. See examples in *Table: Example of interpreting the begin and end angles of the arc*.

Table 8: **Example of interpreting the begin and end angles of the arc**

bgn = 0.0° end = 90.0°	bgn = 90.0° end = 0.0°
	

See also

- *OE2DPoint* class
- *OEPen* class

DrawCircle

```
void DrawCircle(const OE2DPoint& center, double radius,
                const OEPen& pen)
```

Draws a circle. See example in *Figure: Example of circle*.

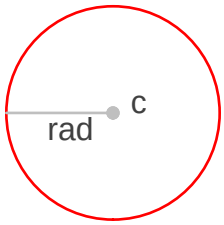


Fig. 133: **Example of circle**

center
The center of the circle.

radius
The radius of the circle.

pen
The graphical properties of the circle (such as color, line width, etc). See examples in *Figure: Example of drawing circles with various pens*.

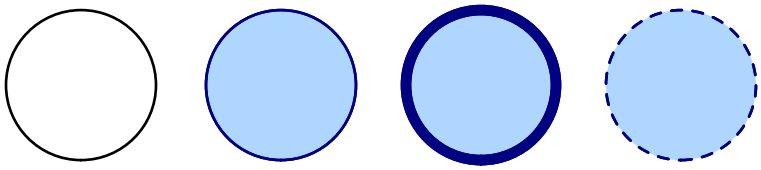


Fig. 134: **Example of drawing circles with various pens**

See also

- *OE2DPoint* class
- *OEPen* class

DrawCubicBezier

```
void DrawCubicBezier(const OE2DPoint& bgn, const OE2DPoint& c1,
                     const OE2DPoint& c2, const OE2DPoint& end,
                     const OEPen& pen)
```

Draws a cubic Bézier curve. See example in *Figure: Example of cubic Bezier curve*.

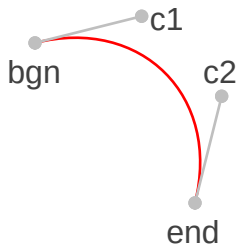


Fig. 135: **Example of cubic Bezier curve**

bgn, end
: The end points of the Bezier curve.

c1, c2
: The control points of the curve.

pen
: The graphical properties of the Bézier curve (such as color, line width, etc). See examples in *Figure: Example of drawing cubic Bezier curves with various pens*.

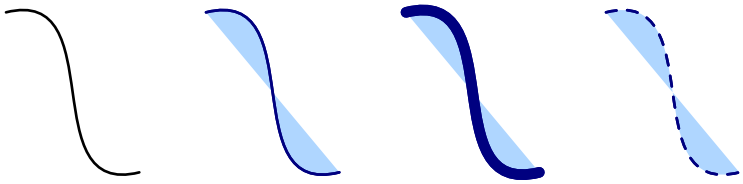


Fig. 136: **Example of drawing cubic Bezier curves with various pens**

Example (*Figure: Example of drawing a cubic Bezier curve*)

```
OEImage image = new OEImage(100, 100);

OE2DPoint b = new OE2DPoint(20, 70);
OE2DPoint e = new OE2DPoint(60, 70);
OE2DPoint c1 = new OE2DPoint(b.GetX() + 50, b.GetY() - 60);
OE2DPoint c2 = new OE2DPoint(e.GetX() + 50, e.GetY() - 60);
```

(continues on next page)

(continued from previous page)

```
OEPen pen = new OEPen(oechem.getOELightGreen(), oechem.getOEBlack(), OEFill.On, 2.0);
image.DrawCubicBezier(b, c1, c2, e, pen);
```

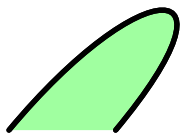


Fig. 137: **Example of drawing a cubic Bezier curve**

See also

- *OE2DPoint* class
- *OEPen* classes

DrawLine

```
void DrawLine(const OE2DPoint& bgn, const OE2DPoint& end,
              const OEPen& pen)
```

Draws a line. See example in *Figure: Example of line*.

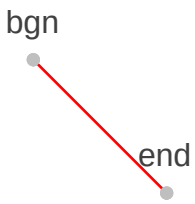


Fig. 138: **Example of line**

bgn*, *end
: The two endpoints of the line.

pen
: The graphical properties of the line (such as color, line width, etc). See examples in *Figure: Example of drawing lines with various pens*.

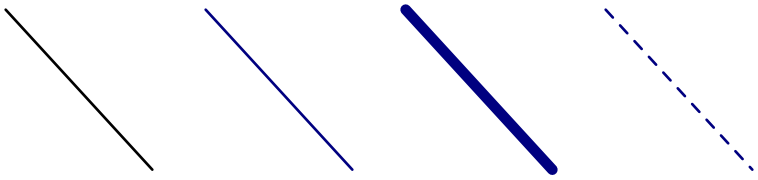


Fig. 139: **Example of drawing lines with various pens**

See also

- *OE2DPoint* class
- *OEPen* class

DrawPath

```
void DrawPath(const OE2DPath& path, const OEPen& pen)
```

Draws as path. See example in *Figure: Example of path*.

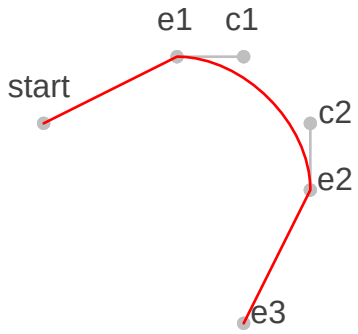


Fig. 140: **Example of path**

path
: The *OE2DPath* object that defines a path as a sequence of lines and curves.

pen
: The graphical properties of the pie (such as color, line width, etc). See examples in *Figure: Example of drawing paths with various pens*.



Fig. 141: **Example of drawing paths with various pens**

Example (*Figure: Example of drawing a path*)

```
OEImage image = new OEImage(100, 100);

OE2DPath path = new OE2DPath(new OE2DPoint(20, 80));
path.AddLineSegment(new OE2DPoint(80, 80));
path.AddLineSegment(new OE2DPoint(80, 40));
path.AddCurveSegment(new OE2DPoint(80, 10), new OE2DPoint(20, 10), new OE2DPoint(20,␣
↪40));

OEPen pen = new OEPen(oechem.getOELightGreen(), oechem.getOEBlack(), OEFill.On, 2.0);
image.DrawPath(path, pen);
```

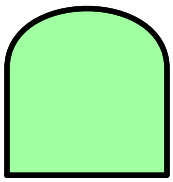


Fig. 142: **Example of drawing a path**

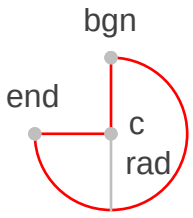
See also

- *OE2DPath* class
- *OEPen* class

DrawPie

```
void DrawPie(const OE2DPoint& center, double bgnAngle,
             double endAngle, double radius, const OEPen& pen)
```

Draws a pie. See example in *Figure: Example of drawing a pie*.



center
The center of the pie.

bgnAngle, endAngle
The two endpoints of the pie (in degrees). Both angles are in degrees and their values have to be in a range of

$[0.0°, 360.0°]$. Angles are interpreted such that $0.0°$ and $360.0°$ degrees are at the *12 o'clock* position, $90.0°$ degrees corresponds to *3 o'clock*, etc.

radius
The radius of the pie.

pen
The graphical properties of the pie (such as color, line width, etc). See examples in *Figure: Example of drawing pies with various pens*.

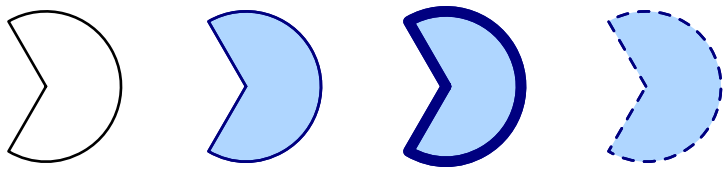
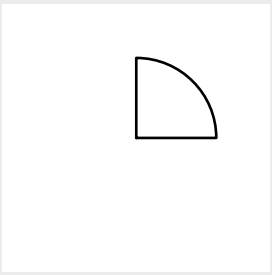
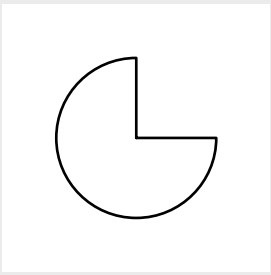


Fig. 143: **Example of drawing pies with various pens**

Pies are always drawn in a clockwise direction. This means that bgn = $0.0°$, end = $90.0°$ angle pair is not the same as bgn = $90.0°$, end = $0.0°$ angle pair. See examples in *Table: Example of interpreting the begin and end angles of the pie*.

Table 9: **Example of interpreting the begin and end angles of the pie**

bgn = $0.0°$ end = $90.0°$	bgn = $90.0°$ end = $0.0°$
	

See also

- *OE2DPoint* class
- *OEPen* class

DrawPoint

```
void DrawPoint(const OE2DPoint& p, const OESystem::OEColor& color) = 0
```

Draws a point. See example in *Figure: Example of point*.

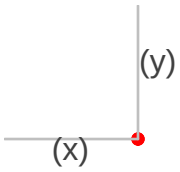


Fig. 144: **Example of point**

p
: The position of the point, where p = OE2DPoint(x, y).

color
: The color of the point.

See also

- OEColor class

DrawPolygon

```
void DrawPolygon(const std::vector<OE2DPoint>& points,
                 const OEPen& pen)
```

Draws a closed polygon. See example in *Figure: Example of polygon*.

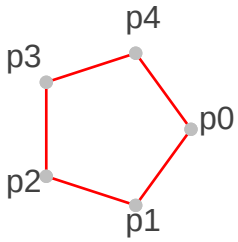


Fig. 145: **Example of polygon**

points
: The series of points that defines the polygon. The polygon will be closed *i.e.* the first connected with the last point.

pen
: The graphical properties of the polygon (such as color, line width, etc). See examples in *Figure: Example of polygons lines with various pens*.

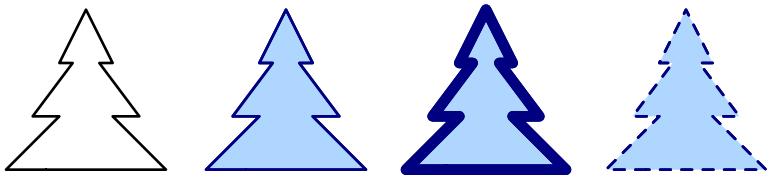


Fig. 146: **Example of drawing polygons with various pens**

Example (*Figure: Example of drawing a polygon*)

```
OEImage image = new OEImage(100, 100);

OE2DPointVector polygon = new OE2DPointVector();

polygon.add(new OE2DPoint(20, 20));
polygon.add(new OE2DPoint(40, 40));
polygon.add(new OE2DPoint(60, 20));
polygon.add(new OE2DPoint(80, 40));
polygon.add(new OE2DPoint(80, 80));
polygon.add(new OE2DPoint(20, 80));

OEPen pen = new OEPen(oechem.getOELightGreen(), oechem.getOEBlack(), OEFill.On, 2.0);
image.DrawPolygon(polygon, pen);
```

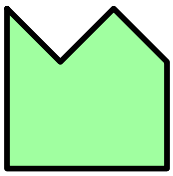


Fig. 147: **Example of drawing a polygon**

See also

- *OE2DPoint* class
- *OEPen* class

DrawQuadraticBezier

```
void DrawQuadraticBezier(const OE2DPoint& bgn, const OE2DPoint& c,
                         const OE2DPoint& end, const OEPen& pen)
```

Draws a quadratic Bézier curve. See example in *Figure: Example of quadratic Bezier curve*.

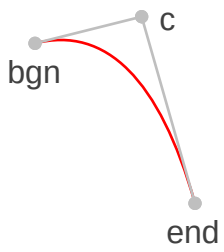


Fig. 148: **Example of quadratic Bezier curve**

bgn, end
The end points of the Bézier curve.

c
The control point of the curve.

pen
The graphical properties of the Bézier curve (such as color, line width, etc). See examples in *Figure: Example of drawing quadratic Bezier curves with various pens*.

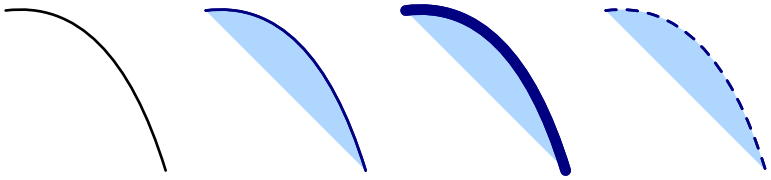


Fig. 149: **Example of drawing quadratic Bezier curves with various pens**

Example (*Figure: Example of drawing a polygon*)

```
OEImage image = new OEImage(100, 100);

OE2DPoint b = new OE2DPoint(20, 70);
OE2DPoint e = new OE2DPoint(80, 70);
OE2DPoint c = new OE2DPoint(b.GetX() + 30, b.GetY() - 80);

OEPen pen = new OEPen(oechem.getOELightGreen(), oechem.getOEBlack(), OEFill.On, 2.0);
image.DrawQuadraticBezier(b, c, e, pen);
```



Fig. 150: **Example of drawing a quadratic Bezier**

See also

- *OE2DPoint* class
- *OEPen* classes

DrawRectangle

```
void DrawRectangle(const OE2DPoint& tl, const OE2DPoint& br,
                   const OEPen& pen)
```

Draws a rectangle. See example in *Figure: Example of rectangle*.

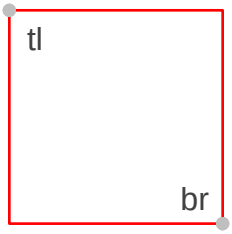


Fig. 151: **Example of rectangle**

tl
: The top left corner of the rectangle.

br
: The bottom right corner of the rectangle.

pen
: The graphical properties of the rectangle (such as color, line width, etc). See examples in *Figure: Example of polygons rectangles with various pens*.

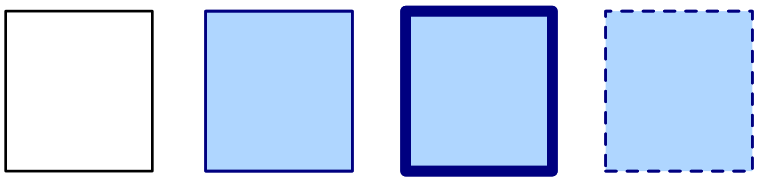


Fig. 152: **Example of drawing rectangles with various pens**

See also

- *OE2DPoint* class
- *OEPen* class

DrawText

```
void DrawText(const OE2DPoint& center, const std::string& text,
              const OEFont& font, double maxwidth = 0.0)
```

Draws a text. See example in *Figure: Example of text*.



Fig. 153: **Example of text**

center
The position of the text.

text
The characters to be displayed.

font
The graphical properties of the displayed text (such as color, font size, etc). See examples in *Figure: Example of drawing texts with various fonts*.

maxwidth
If *maxwidth* is specified and the width of the text would exceed the given limit, than the size of the font used to display the text is automatically reduced.

Note

The displayed position of the text not only depends on 'c' position but also the alignment style of the given *OEFont*.



Fig. 154: **Example of drawing texts with various fonts**

See also

- *OE2DPoint* class
- *OEAlignment* namespace
- *OEFont* class

DrawTriangle

```
void DrawTriangle(const OE2DPoint& a, const OE2DPoint& b,
                  const OE2DPoint& c, const OEPen& pen)
```

Draws a triangle. See example in *Figure: Example of triangle*.

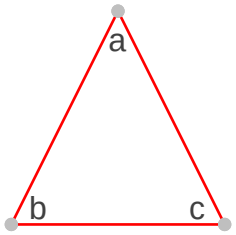


Fig. 155: **Example of triangle**

a*, *b*, *c
The three corners of the triangle.

pen
The graphical properties of the triangle (such as color, line width, etc). See examples in *Figure: Example of polygons triangles with various pens*.

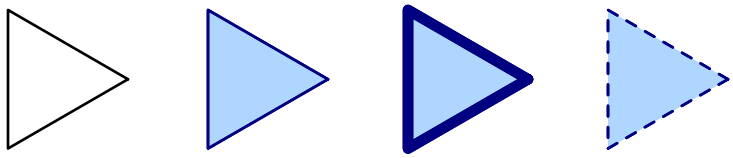


Fig. 156: **Example of drawing triangles with various fonts**

See also

- *OE2DPoint* class
- *OEPen* class

GetGlobalOffset

```
const OE2DPoint& GetGlobalOffset() const
```

Returns the offset from the origin (x=0.0, y=0.0) to the top-left corner of the given image.

GetHeight

```
double GetHeight() const
```

Returns the height of the *OEImageBase* object.

GetMinFontSize

```
unsigned int GetMinFontSize() const
```

Returns the minimum size of the font that can be displayed.

GetWidth

```
double GetWidth() const
```

Returns the width of the *OEImageBase* object.

SetMinFontSize

```
void SetMinFontSize(unsigned int fontsize)
```

Sets the minimum size of the font that can be displayed.

GetSVGClass

```
OESVGClass* GetSVGClass(const std::string& name)
const OESVGClass* GetSVGClass(const std::string& name) const
```

Returns the *OESVGClass* object with the given name. Null pointer will be returned if no class exists on the image with the given name.

See also

- *Generating Interactive SVG Images* chapter
- *OESVGClass* class

GetSVGGroup

```
OESVGGroup* GetSVGGroup(const std::string& id)
const OESVGGroup* GetSVGGroup(const std::string& id) const
```

Returns the *OESVGGroup* object with the given name. Null pointer will be returned if no group exists on the image with the given name.

See also

- *Generating Interactive SVG Images* chapter
- *OESVGGroup* class

GetSVGGroups

```
OESystem::OEIterBase<OESVGGroup>* GetSVGGroups()
```

Returns an iterator over the SVG groups (*OESVGGroup*) of the image.

Example:

```
for (OESVGGroup group : image.GetSVGGroups())
    System.out.println(group.GetId());
```

See also

- *Generating Interactive SVG Images* chapter
- *OESVGGroup* class

NewSVGClass

```
OESVGClass* NewSVGClass(const std::string& name)
```

Creates a new *OESVGClass* on the given image. A null pointer will be returned and warning message will be thrown if the SVG class can not be created.

name
: The name of the *OESVGClass* object. For information about valid group names see the documentation of the *OEIsValidSVGClassName* function.

See also

- *Generating Interactive SVG Images* chapter
- *OESVGClass* class

NewSVGGroup

```
OESVGGroup* NewSVGGroup(const std::string& id)
```

Creates a new *OESVGGroup* on the given image. A null pointer will be returned and warning message will be thrown if the SVG group can not be created.

name
: The name of the *OESVGGroup* object. For information about valid group names see the documentation of the *OEIsValidSVGGroupId* function.

See also

- *Generating Interactive SVG Images* chapter
- *OESVGGroup* class

PopGroup

```
bool PopGroup(const OESVGGroup *)
```

Pops the *OESVGGroup* object.

Note

An *OESVGGroup* object can be used (pushed / popped) only once. *OESVGGroup* objects can be nested. They should be used in LIFO (for last in, first out) i.e. the group that has been pushed last should be popped first.

See also

- *Generating Interactive SVG Images* chapter
- *OESVGGroup* class
- *OEImageBase.PushGroup* method

PushGroup

```
bool PushGroup(const OESVGGroup *)
```

Pushes the *OESVGGroup* object.

Note

An *OESVGGroup* object can be used (pushed / popped) only once. *OESVGGroup* objects can be nested. They should be used in LIFO (for last in, first out) i.e. the group that has been pushed last should be popped first.

See also

- *Generating Interactive SVG Images* chapter
- *OESVGGroup* class
- *OEImageBase.PopGroup* method

4.1.38 OEImageFileBase

```
class OEImageFileBase : public OEImageBase
```

The *OEImageFileBase* is an abstract class that is used to write concrete image files.

The following methods are publicly inherited from *OEImageBase*:

Clear	*DrawPoint*	*GetHeight*
DrawArc	*DrawPolygon*	*GetMinFontSize*
DrawCircle	*DrawQuadraticBezier*	*GetWidth*
DrawCubicBezier	*DrawRectangle*	*SetMinFontSize*
DrawLine	*DrawText*	
DrawPie	*DrawTriangle*	

Constructors

```
OEImageFileBase(double width, double height)
```

Default constructors that creates an image file with the given dimensions.

GetExtension

```
std::string GetExtension() const = 0
```

Returns the file extension associated with the concrete image file type.

IsVisible

```
virtual bool IsVisible() const
```

Returns whether the top active SVG group is visible i.e. returns *OESVGGroup.IsVisible*.

PopGroup

```
virtual bool PopGroup (const OESVGGroup*)
```

Pops the *OESVGGroup* object.

See also

- *Generating Interactive SVG Images* chapter
- *OESVGGroup* class
- *OEImageFileBase.PushGroup* method

PushGroup

```
virtual bool PushGroup(const OESVGGroup*)
```

Pushes the *OESVGGroup* object.

See also

- *Generating Interactive SVG Images* chapter
- *OESVGGroup* class
- *OEImageFileBase.PopGroup* method

Write

```
bool Write(OEPlatform::oeostream& os)=0
```

The *OEImageFileBase.Write* is a virtual method that has to be implemented in the concrete derived classes and it writes the image into the given output stream.

4.1.39 OEImageFileCreatorBase

```
class OEImageFileCreatorBase
```

The *OEImageFileCreatorBase* is an abstract class that is used to register concrete image types.

See also

- *OERegisterImageFile* function

Constructors

```
OEImageFileCreatorBase(const std::string& ext, bool multi = false)
```

Default constructor.

ext
: The file extension that will be used to identify the concrete image type. For example, the Portable Document Format (PDF) is registered with the "pdf" file extension.

multi
: Determines whether the specific image type can be used to create multi-page images.

CreateCopy

```
OEImageFileCreatorBase* CreateCopy() const = 0
```

Virtual const constructor which allows copying of concrete derived objects using a reference to this base class.

CreateImage

```
OEImageFileBase* CreateImage(double width, double height) const = 0
```

The *OEImageFileCreatorBase.CreateImage* is a virtual method that has to be implemented in the concrete derived classes and it returns a pointer of concrete image with the given dimensions.

width
: The width of the returned image.

height
: The height of the returned image.

GetExtension

```
const std::string GetExtension() const
```

Return the file extension of which the concrete image type is being registered.

IsMultiPage

```
bool IsMultiPage() const
```

Returns whether the image type registered as multi-page image.

4.1.40 OEImageFrame

```
class OEImageFrame : public OEImageBase
```

OEImageFrame class provides a convenient way to create a *frame* with a specific dimension and offset relative to an *OEImageBase* object. See *Figure: Schematic representation of OEImageFrame objects*.

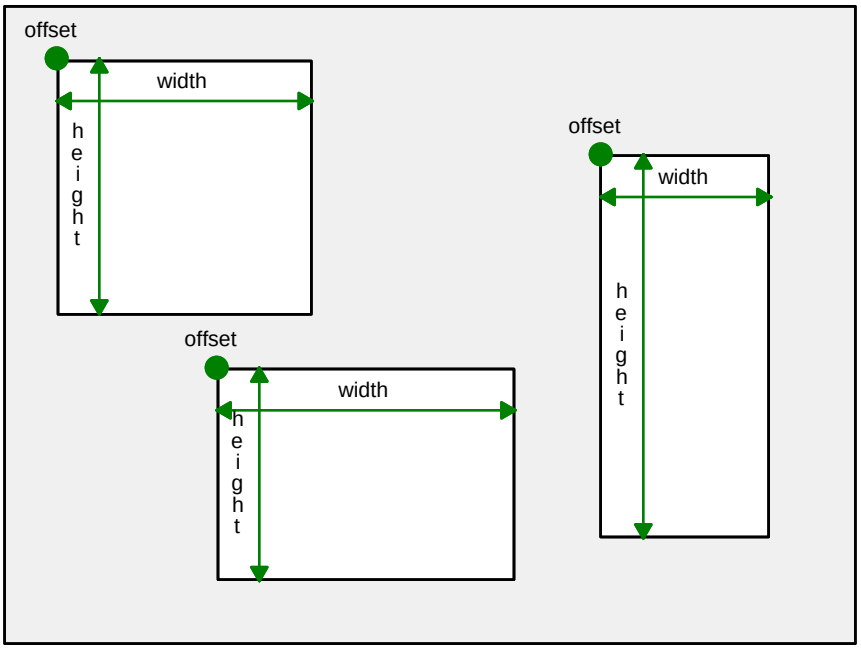


Fig. 157: **Schematic representation of OEImageFrame objects**

See also

- *Depicting Molecules in an Arbitrary Position* section

The following methods are publicly inherited from *OEImageBase*:

Clear	*DrawPie*	*DrawTriangle*
DrawArc	*DrawPoint*	*GetGlobalOffset*
DrawCircle	*DrawPolygon*	*GetHeight*
DrawCubicBezier	*DrawQuadraticBezier*	*GetMinFontSize*
DrawLine	*DrawRectangle*	*GetWidth*
DrawPath	*DrawText*	*GetSVGClass*
GetSVGGroup	*GetSVGGroups*	*NewSVGClass*
NewSVGGroup	*PopGroup*	*PushGroup*
SetMinFontSize		

Constructors

```
OEImageFrame(OEImageBase& parent, double width, double height,
             const OE2DPoint& offset)
```

Creates an *OEImageFrame* object with the given dimension.

parent
: The parent *OEImageBase* object in which the *OEImageFrame* object is placed.

width
: The width of the *OEImageFrame* object.

height
: The height of the *OEImageFrame* object.

offset
: The offset of the *OEImageFrame* object relative to top/left corner of its *OEImageBase* parent.

```
OEImageFrame(OEImageBase& parent, double padding)
```

Creates an *OEImageFrame* object equal paddings around the given image positioning the image frame to the center of the image.

parent
: The parent *OEImageBase* object in which the *OEImageFrame* object is placed.

padding
: The padding that applied in all four sides around the parent image. The padding is capped by the $\frac{1}{3}$ of width and height of the **parent** image.

Clear

```
void Clear(const OESystem::OEColor& color)
```

Clears the image with the given color.

See also

- *OEImageBase.Clear* method
- OEColor class

DrawArc

```
void DrawArc(const OE2DPoint& center, double bgnAngle,
             double endAngle, double radius, const OEPen& pen)
```

Draws the outline of an arc. See example in *Figure: Example of drawing an arc*.

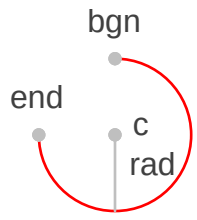


Fig. 158: **Example of drawing an arc**

See also

- *OEImageBase.DrawArc* method
- *OE2DPoint* class

- *OEPen* class

DrawCircle

```
void DrawCircle(const OE2DPoint& center, double radius,
                const OEPen& pen)
```

Draws a circle. See example in *Figure: Example of drawing a circle*.

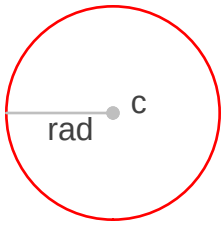


Fig. 159: **Example of drawing a circle**

See also

- *OEImageBase.DrawCircle* method
- *OE2DPoint* class
- *OEPen* class

DrawCubicBezier

```
void DrawCubicBezier(const OE2DPoint& bgn, const OE2DPoint& c1,
                     const OE2DPoint& c2, const OE2DPoint& end,
                     const OEPen& pen)
```

Draws a cubic Bézier curve. See example in *Figure: Example of drawing a cubic Bezier curve*.

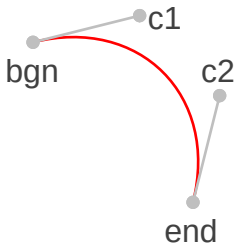


Fig. 160: **Example of drawing a cubic Bezier curve**

See also

- *OEImageBase.DrawCubicBezier* method
- *OE2DPoint* class
- *OEPen* classes

DrawLine

```
void DrawLine(const OE2DPoint& bgn, const OE2DPoint& end,
              const OEPen& pen)
```

Draws a line. See example in *Figure: Example of drawing a line*.

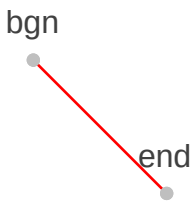


Fig. 161: **Example of drawing a line**

See also

- *OEImageBase.DrawLine* method
- *OE2DPoint* class
- *OEPen* classes

DrawPath

```
void DrawPath(const OE2DPath& path, const OEPen& pen)
```

Draws a path. See example in *Figure: Example of drawing a path*.

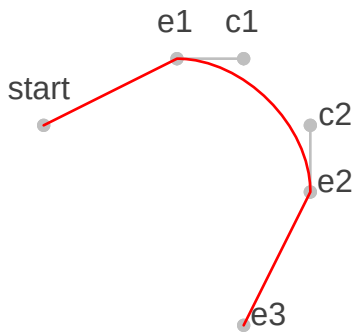


Fig. 162: **Example of drawing a path**

See also

- *OEImageBase.DrawPath* methods
- *OE2DPath* class
- *OEPen* classes

DrawPie

```
void DrawPie(const OE2DPoint& center, double bgnAngle,
             double endAngle, double radius, const OEPen& pen)
```

Draws a pie. See example in *Figure: Example of drawing a pie*.

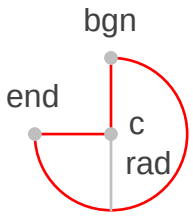


Fig. 163: **Example of drawing a pie**

See also

- *OEImageBase.DrawPie* methods
- *OE2DPoint* class
- *OEPen* classes

DrawPoint

```
void DrawPoint(const OE2DPoint& p, const OESystem::OEColor& color)
```

Draws a point. See example in *Figure: Example of drawing a point*.

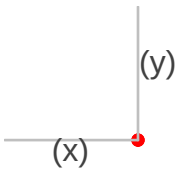


Fig. 164: **Example of drawing a point**

See also

- *OEImageBase.DrawPoint* method
- *OE2DPoint* class
- OEColor class

DrawPolygon

```
void DrawPolygon(const std::vector<OE2DPoint>& points,
                 const OEPen& pen)
```

Draws a closed polygon. See example in *Figure: Example of drawing a polygon*.

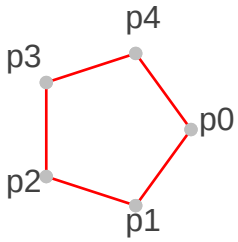


Fig. 165: **Example of drawing a polygon**

See also

- *OEImageBase.DrawPolygon* method
- *OE2DPoint* class
- *OEPen* class

DrawQuadraticBezier

```
void DrawQuadraticBezier(const OE2DPoint& bgn, const OE2DPoint& c,
                         const OE2DPoint& end, const OEPen& pen)
```

Draws a quadratic Bézier curve. See example in *Figure: Example of drawing a quadratic Bezier curve*.

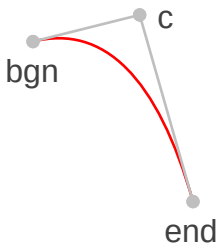


Fig. 166: **Example of drawing a quadratic Bezier curve**

See also

- *OEImageBase.DrawQuadraticBezier* methods
- *OE2DPoint* class
- *OEPen* classes

DrawRectangle

```
void DrawRectangle(const OE2DPoint& tl, const OE2DPoint& br,
                   const OEPen& pen)
```

Draws a rectangle. See example in *Figure: Example of drawing a rectangle*.

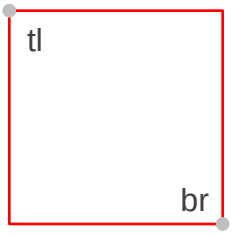


Fig. 167: **Example of drawing a rectangle**

See also

- *OEImageBase.DrawRectangle* method
- *OE2DPoint* class
- *OEPen* class

DrawText

```
void DrawText(const OE2DPoint& c, const std::string& text,
              const OEFont& font, double maxwidth = 0.0)
```

Draws a text. See example in *Figure: Example of drawing a text*.



Fig. 168: **Example of drawing a text**

See also

- *OEImageBase.DrawText* method
- *OE2DPoint* class
- *OEFont* class

DrawTriangle

```
void DrawTriangle(const OE2DPoint& a, const OE2DPoint& b,
                  const OE2DPoint& c, const OEPen& pen)
```

Draws a triangle. See example in *Figure: Example of drawing a triangle*.

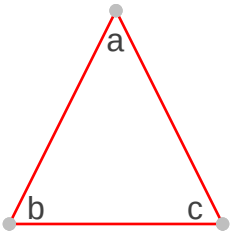


Fig. 169: **Example of drawing a triangle**

See also

- *OEImageBase.DrawTriangle* method
- *OE2DPoint* class
- *OEPen* classes

GetHeight

```
double GetHeight() const
```

Returns the height of the *OEImageFrame* object.

GetOffset

```
const OE2DPoint& GetOffset() const
```

Returns the offset of the *OEImageFrame* object relative to its *OEImageBase* parent.

GetWidth

```
double GetWidth() const
```

Returns the width of the *OEImageFrame* object.

4.1.41 OEImageGrid

```
class OEImageGrid
```

This class represents *OEImageGrid* which is a layout manager that aligns cells into rows and columns. The cells of the grid are equal-sized and evenly spaced. See *Figure: Schematic representation of an OEImageGrid object*.

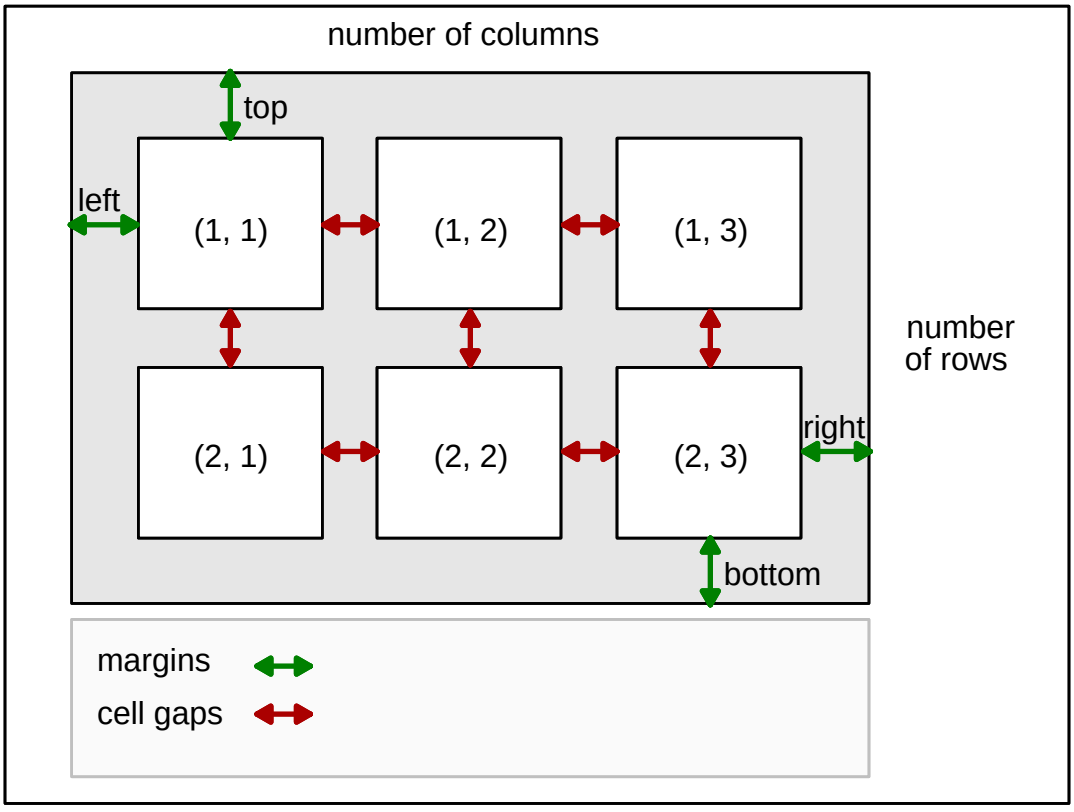


Fig. 170: **Schematic representation of an OEImageGrid object**

Note

In the image grid, each cell has the same width and height. The width and height of the cells are calculated based on the height and width of the *OEImageGrid*, its margins and gap set between the cells.

See also

- *Depicting Molecules in a Grid* section
- *OEImageTable* class

Constructors

```
OEImageGrid(OEImageBase& parent, unsigned int rows,
            unsigned int cols)
```

Creates an *OEImageGrid* object with the specified number of rows and columns.

parent
The *OEImageBase* object on which the *OEImageGrid* object is placed.

rows
The number of rows in the *OEImageGrid* object.

cols
The number of columns in the *OEImageGrid* object.

Note

The width and the height of the *OEImageGrid* object are identical to the width and height of the *parent* *OEImageBase*, respectively.

See also

- *OEImageGrid.NumCols* method
- *OEImageGrid.NumRows* method

GetCell

```
OEImageBase* GetCell(unsigned int r, unsigned int c)
```

Returns the *OEImageBase* pointer for the cell positioned in the r^{th} row and c^{th} columns.

r
This value has to be in the range from *1* to *OEImageGrid.NumRows()*.

c
This value has to be in the range from *1* to *OEImageGrid.NumCols()*.

GetCellGap

```
double GetCellGap() const
```

Returns the space between the cells of the *OEImageGrid* object.

See also

- *OEImageGrid.SetCellGap* method

GetCellHeight

```
double GetCellHeight() const
```

Returns the height of the cells of the *OEImageGrid* object.

GetCellWidth

```
double GetCellWidth() const
```

Returns the width of the cells of the *OEImageGrid* object.

GetCells

```
OESystem::OEIterBase<OEImageBase>* GetCells(unsigned int row = 0u,
                                            unsigned int col = 0u)
```

Returns an iterator over the cells of the *OEImageGrid* object. The cells are returned from left to right and top to bottom order.

row
: This value has to be in the range from *0* to *OEImageGrid.NumRows()*.

col
: This value has to be in the range from *0* to *OEImageGrid.NumCols()*.

Example (*Figure Example of accessing all cells of the grid*)

```
for (OEImageBase cell : grid.GetCells()) {
```

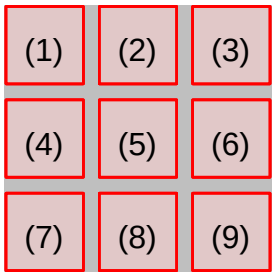


Fig. 171: **Example of accessing all cells of the grid.**
(The numbers inside the cells represent the order in which the cells are returned)

If the *col* number is zero then *OEImageGrid.GetCells* returns an iterator over the cells of the specified row.

Example (*Figure Example of accessing a specific row of the grid*)

```
int row = 2;
for (OEImageBase cell : grid.GetCells(row, 0)) {
```

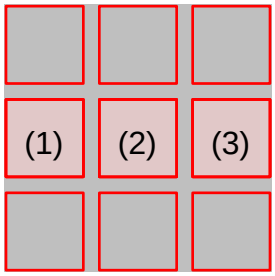


Fig. 172: **Example of accessing a specific row of the grid.**
(The numbers inside the cells represent the order in which the cells are returned)

If the *row* number is zero then *OEImageGrid.GetCells* returns an iterator over the cells of the specified column.

Example (*Figure Example of accessing a specific column of the grid*)

```
int col = 2;
for (OEImageBase cell : grid.GetCells(0, col)) {
```

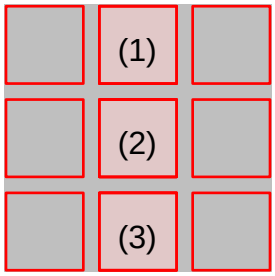


Fig. 173: **Example of accessing a specific column of the grid.**
(The numbers inside the cells represent the order in which the cells are returned)

GetMargin

```
double GetMargin(unsigned int margin) const
```

Returns the size of a specific margin of the *OEImageGrid* object.

margin
This value has to be from the *OEMargin* namespace.

See also

- *OEImageGrid.SetMargin* method
- *OEImageGrid.SetMargins* method

NumCols

```
unsigned int NumCols() const
```

Returns the number of columns of the *OEImageGrid* object.

NumRows

```
unsigned int NumRows() const
```

Returns the number of rows of the *OEImageGrid* object.

SetCellGap

```
bool SetCellGap(double size)
```

Sets the space between the cells of the *OEImageGrid* object to the given size.

size
: This value has to be a positive number.

See also

- *OEImageGrid.GetCellGap* method

SetMargin

```
bool SetMargin(unsigned int margin, double size)
```

Sets the size of a specific margin of the *OEImageGrid* object.

margin
: This value has to be from the *OEMargin* namespace.

size
: This value has to be a positive number.

See also

- *OEImageGrid.GetMargin* method

SetMargins

```
bool SetMargins(double size)
```

Sets the size of all margins of the *OEImageGrid* object.

size
: This value has to be a positive number.

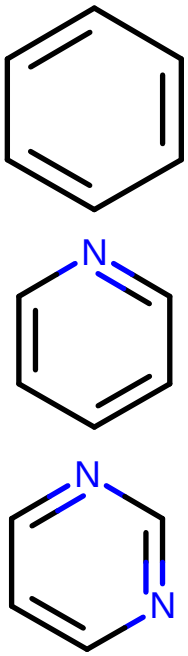
See also

- *OEImageGrid.GetMargin* method

4.1.42 OEImageTable

```
class OEImageTable
```

This class represents *OEImageTable* which is a layout manager that allows to arrange information in rows and columns with variable height and width, respectively.



Property	Value
Name	benzene
SMILES	c1ccccc1
MV	78.112

Property	Value
Name	pyridine
SMILES	c1ccncc1
MV	79.100

Property	Value
Name	pyrimidine
SMILES	c1cncnc1
MV	80.088

Fig. 174: **Example of using image table**

See also

- *Depicting Molecules in a Table* section
- *OEImageTableOptions* class
- *OEImageTableStyle* namespace
- *OEImageGrid* class

Constructors

```
OEImageTable(OEImageBase& parent, const OEImageTableOptions& opts)
```

Creates an *OEImageTable* object with the specified options.

parent
The *OEImageBase* object on which the *OEImageTable* object is placed.

opts
The *OEImageTableOptions* object that defines the properties and style of the table.

See also

- *OEImageTable.GetOptions* method
- *OEImageTable.NumColumns* method
- *OEImageTable.NumRows* method

DrawText

```
bool DrawText(OEImageBase* cell, const std::string& text,
              double padding = 5.0)
```

Draws a text to the given table cell. Returns `false` if the given image does not belong to the table.

padding
Defines the padding used on the left and right of the given cell as a percentage of the width of the cell.

Note

The font that is used to render the text is defined in the *OEImageTableOptions* class.

See also

- *OEFont* class
- *OEImageTableOptions.SetCellFont* method
- *OEImageTableOptions.SetHeaderFont* method
- *OEImageTableOptions.SetStubColumnFont* method

GetBodyCell

```
OEImageBase* GetBodyCell(unsigned int row, unsigned int col)
```

Returns the *OEImageBase* for the cell positioned in the r^{th} row and c^{th} columns of the 'body' of the table. A cell is considered to be part of the 'body' of the table if it is neither in the header nor in the stub column.

row
This value has to be in the range from 1 to *OEImageTable.NumRows(true)*.

col
This value has to be in the range from 1 to *OEImageTable.NumColumns(true)*.

Note

A null pointer is returned if the **row** or the **col** parameters are outside the acceptable range.

Example

The image below marks the cell returned by *OEImageTable.GetBodyCell* method for row = 1 and col = 2 parameters.

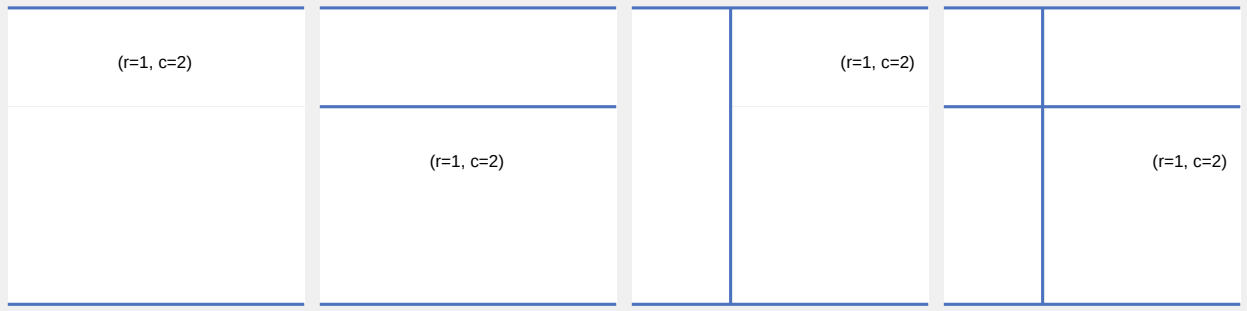


Fig. 175: **Example of accessing a specific 'body' cell of the table.**

See also

- *OEImageTable.GetCell* method

GetBodyCells

```
OESystem::OEIterBase<OEImageBase>* GetBodyCells(unsigned int row = 0u,
                                                unsigned int col = 0u)
```

Returns an iterator over the 'body' cells of the *OEImageTable* object. A cell is considered to be part of the 'body' of the table if it is neither in the header nor in the stub column. The cells are returned from left to right and top to bottom order.

row
: This value has to be in the range from 0 to *OEImageTable.NumRows(true)*.

col
: This value has to be in the range from 0 to *OEImageTable.NumColumns(true)*.

Note

An empty iterator is returned if the **row** or the **col** parameters are outside the acceptable range.

Examples

If the *OEImageTable.GetBodyCells* method is invoked without any parameter, then all 'body cells' are returned.

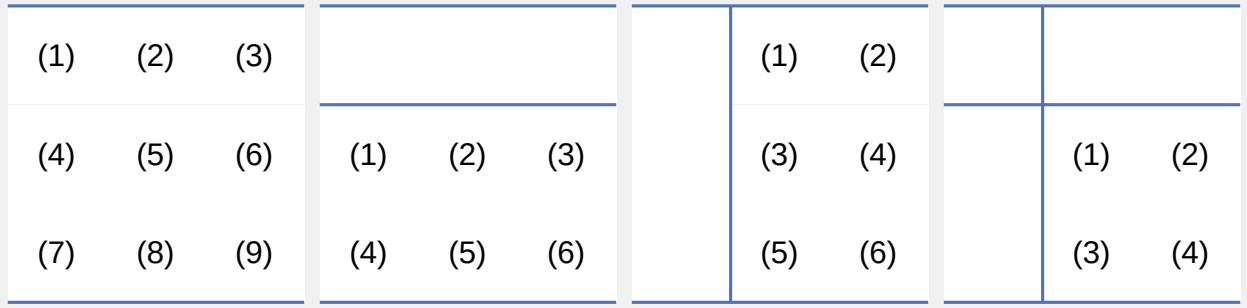


Fig. 176: **Example of accessing all 'body' cells of the table**
(The numbers inside the cells represent the order in which the cells are returned)

If the **row** parameter is zero, then the *OEImageTable.GetBodyCells* method returns an iterator over the 'body' cells of the specified column. The image below marks the cells returned by *OEImageTable.GetBodyCell* method for row = 0 and col = 2 parameters.

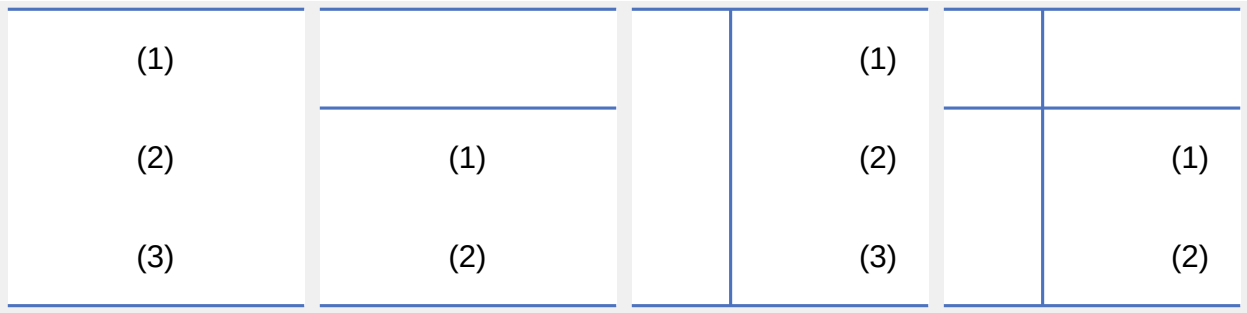


Fig. 177: **Example of accessing 'body' cells of a specific columns in the table**
(The numbers inside the cells represent the order in which the cells are returned)

If the **column** parameter is zero, then the *OEImageTable.GetBodyCells* method returns an iterator over the 'body' cells of the specified column. The image below marks the cells returned by *OEImageTable.GetBodyCell* method for row = 2 and col = 0 parameters.

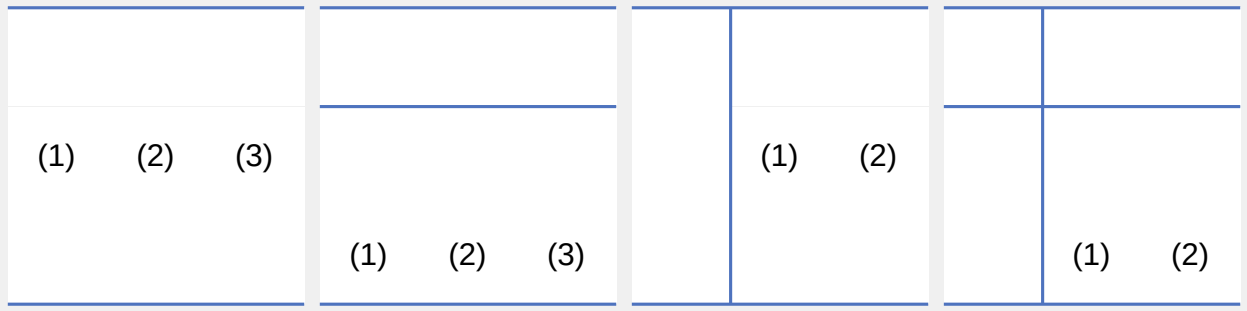


Fig. 178: **Example of accessing 'body' cells of a specific row in the table**
(The numbers inside the cells represent the order in which the cells are returned)

See also

- *OEImageTable.GetCells* method

GetCell

```
OEImageBase* GetCell(unsigned int row, unsigned int col)
```

Returns the *OEImageBase* pointer for the cell positioned in the r^{th} row and c^{th} columns.

row
: This value has to be in the range from 1 to *OEImageTable.NumRows()*.

col
: This value has to be in the range from 1 to *OEImageTable.NumColumns()*.

Note

A null pointer is returned if the **row** or the **col** parameters are outside the acceptable range.

Example

The image below marks the cell returned by *OEImageTable.GetCell* method for row = 1 and col = 2 parameters.

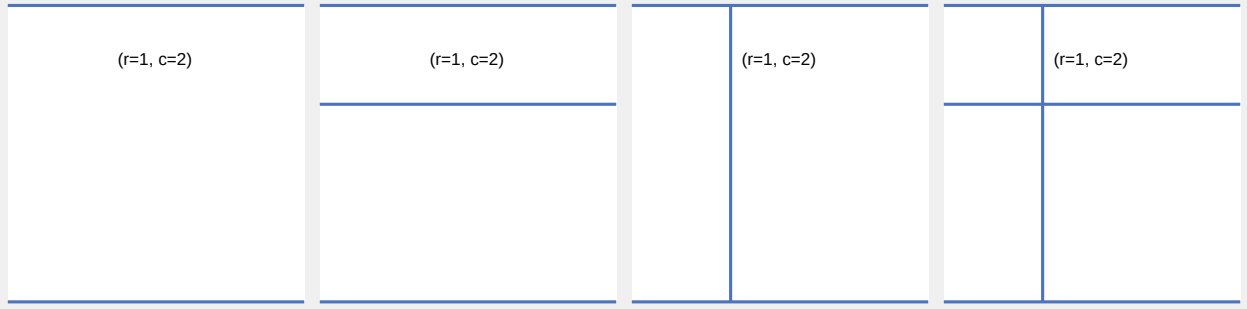


Fig. 179: **Example of accessing a specific cell of the table**

See also

- *OEImageTable.GetBodyCell* method

GetCells

```
OESystem::OEIterBase<OEImageBase>* GetCells (unsigned int row = 0u,
                                             unsigned int col = 0u)
```

Returns an iterator over the cells of the *OEImageTable* object. The cells are returned from left to right and top to bottom order.

row
: This value has to be in the range from 0 to *OEImageTable.NumRows()*.

col
: This value has to be in the range from 0 to *OEImageTable.NumColumns()*.

Note

An empty iterator is returned if the **row** or the **col** parameters are outside the acceptable range.

Examples

If the *OEImageTable.GetCells* method is invoked without any parameter, then all cells of the table are returned.

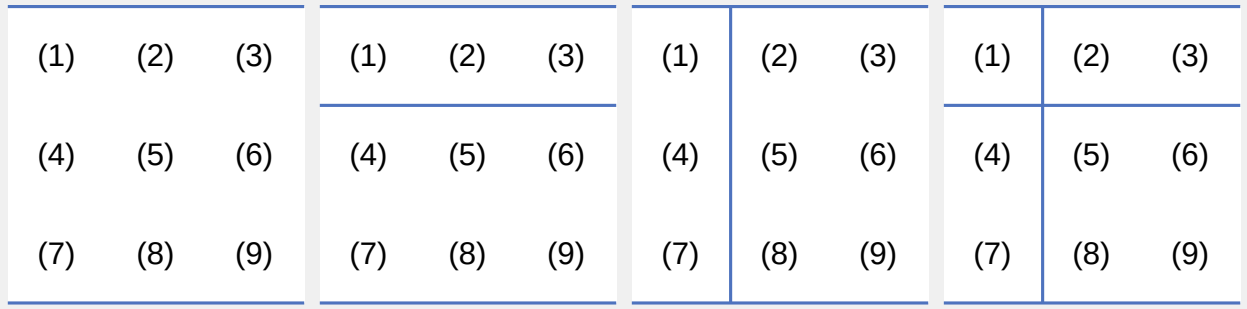


Fig. 180: **Example of accessing all cells of the table**
(The numbers inside the cells represent the order in which the cells are returned)

If the **row** parameter is zero, then the *OEImageTable.GetCells* method returns an iterator over the cells of the specified column. The image below marks the cells returned by *OEImageTable.GetCell* method for row = 0 and col = 2 parameters.

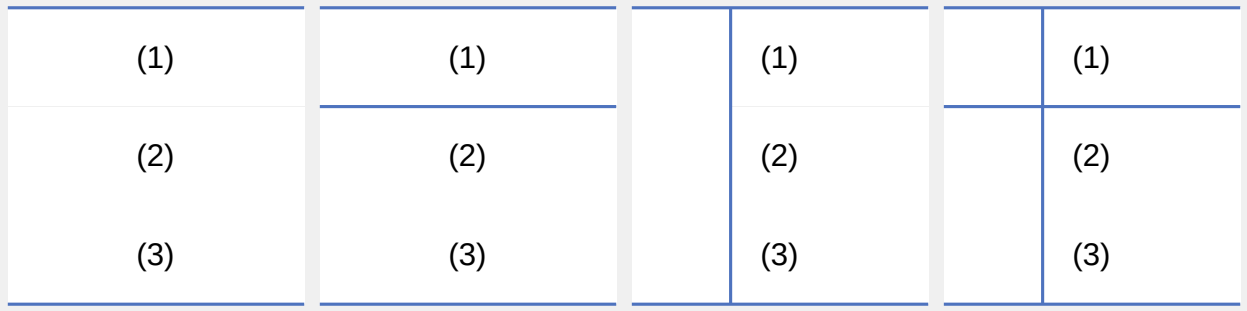


Fig. 181: **Example of accessing all cells of a specific columns of the table.**
(The numbers inside the cells represent the order in which the cells are returned)

If the **column** parameter is zero, then the *OEImageTable.GetCells* method returns an iterator over the cells of the specified row. The image below marks the cells returned by *OEImageTable.GetCell* method for row = 2 and col = 0 parameters.

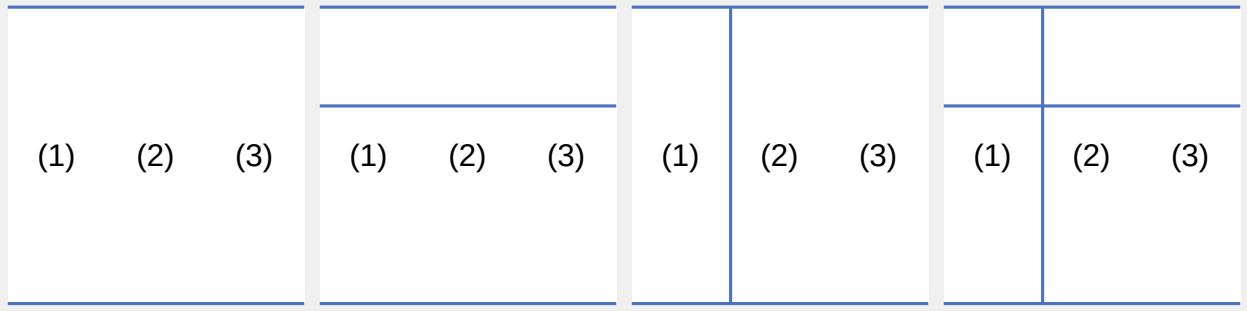


Fig. 182: **Example of accessing all cells of a specific row of the table.**
(The numbers inside the cells represent the order in which the cells are returned)

See also

- *OEImageTable.GetBodyCells* method

GetHeaderCell

```
OEImageBase* GetHeaderCell(unsigned int col, bool skipstub = true)
```

Returns the *OEImageBase* for the cell positioned in the c^{th} columns of the 'header' of the table.

col
This value has to be in the range:

- from 1 to *OEImageTable.NumColumns(false)* if **skipstub** is true.
- from 1 to *OEImageTable.NumColumns(true)* if **skipstub** is false.

skipstub
The parameter that determines whether to consider the stub column. If **skipstub** is *false* and the table has a stub column, than that stub column is considered as the first column.

Note

A null pointer is returned if the table has no header or if the **col** parameters is outside the acceptable range.

Example

The image below marks the cell returned by *OEImageTable.GetHeaderCell* method for col = 2 and skipstub = true parameters.

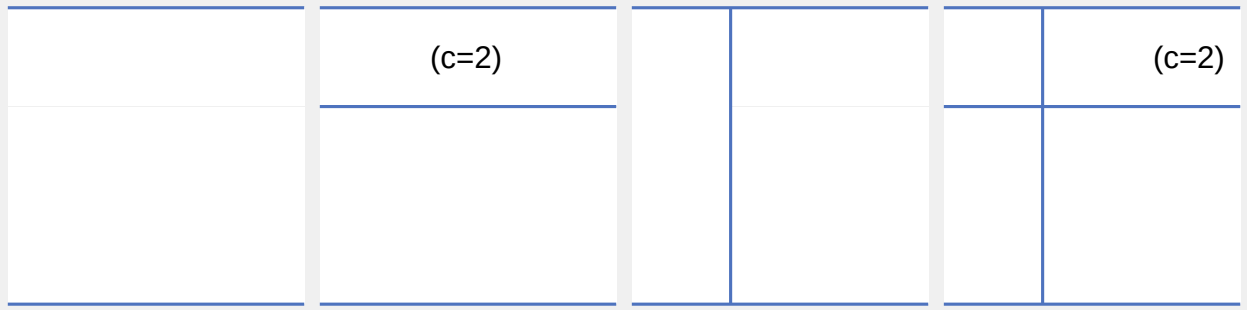


Fig. 183: **Example of accessing a specific header cell of the table**

GetHeaderCells

```
OESystem::OEIterBase<OEImageBase>* GetHeaderCells(bool skipstub = true)
```

Returns an iterator over the header cells of the *OEImageTable* object. The cells are returned from left to right order.

skipstub
: The parameter that determines whether to consider the stub column. If **skipstub** is *false* and the table has a stub column, than that stub column is considered as the first column.

Note

An empty iterator is returned if table has no header row.

Example

The image below marks the cells returned by *OEImageTable.GetHeaderCells* for skipstub = true parameter.

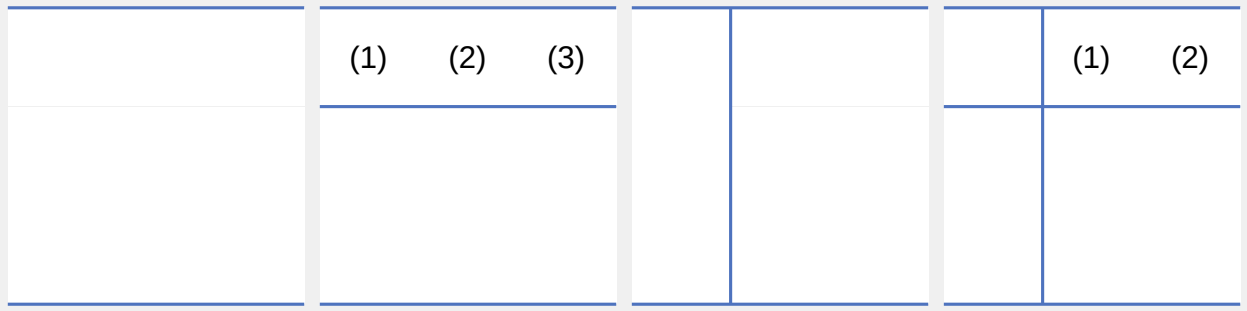


Fig. 184: **Example of accessing header cells of the table**
(The numbers inside the cells represent the order in which the cells are returned)

GetOptions

```
const OEImageTableOptions& GetOptions() const
```

Returns the options used to initialized the *OEImageTable* object.

See also

- *OEImageTableOptions* class

GetStubColumnCell

```
OEImageBase* GetStubColumnCell(unsigned int row)
```

Returns the *OEImageBase* for the cell positioned in the r^{th} row of the 'stub column' of the table.

row
This value has to be in the range from 1 to *OEImageTable.NumRows(true)*.

Note

A null pointer is returned if the table has no stub column or if the **row** parameters is outside the acceptable range.

Example

The image below marks the cell returned by *OEImageTable.GetStubColumnCell* method for row=1 parameter.

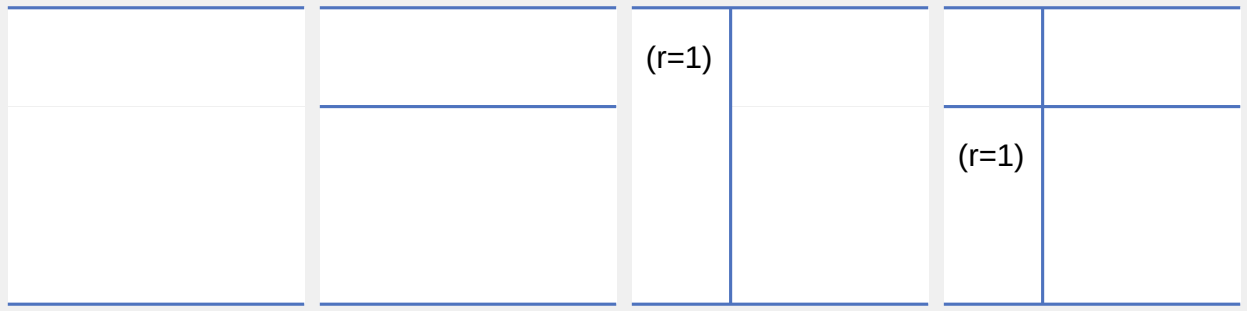


Fig. 185: **Example of accessing a specific stub column cell of the table**

GetStubColumnCells

```
OESystem::OEIterBase<OEImageBase>* GetStubColumnCells()
```

Returns an iterator over the stub column cells of the *OEImageTable* object. The cells are returned from top to bottom order.

Note

An empty iterator is returned if table has no stub column.

Example

The image below marks the cells returned by *OEImageTable.GetStubColumnCells* method.

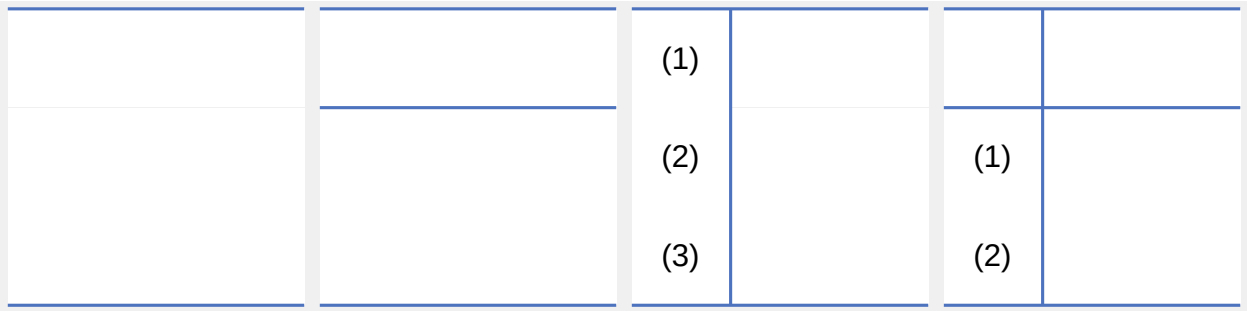


Fig. 186: **Example of accessing stub column cells of the table**
(The numbers inside the cells represent the order in which the cells are returned)

NumColumns

```
unsigned int NumColumns(bool onlybody = false) const
```

Returns the number of columns of the *OEImageTable* object.

onlybody
If this parameter is 'true', than stub column is not counted if exists.

NumRows

```
unsigned int NumRows(bool onlybody = false) const
```

Returns the number of rows of the *OEImageTable* object.

onlybody
: If this parameter is 'true', than header row is not counted if exists.

4.1.43 OEImageTableOptions

```
class OEImageTableOptions
```

This class represents the *OEImageTableOptions* class that encapsulates properties that determine how an image table is depicted.

The *OEImageTableOptions* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
number of columns	*NumColumns*		integer greater than 0
number of rows	*NumRows*		integer greater than 1
width of columns		*SetColumnWidths*	vector of non zero integers
height of rows		*SetRowHeights*	vector of non zero integers
cell font size		*SetBaseFontSize*	positive non-zero integer
pen used around cells cells	*GetCellBorderPen*	*SetCellBorderPen*	*OEPen*
table cell color	*GetCellColor*	*SetCellColor*	OEColor
table cell font	*GetCellFont*	*SetCellFont*	*OEFont*
header	*HasHeader*	*SetHeader*	boolean
header color	*GetHeaderColor*	*SetHeaderColor*	OEColor
header font	*GetHeaderFont*	*SetHeaderFont*	*OEFont*
table margins	*GetMargin*	*SetMargin* / *SetMargins*	positive floating point number in the range of $[0.0, 25.0]$
stub column	*HasStubColumn*	*SetStubColumn*	boolean
stub column color	*GetStubColumnColor*	*SetStubColumnColor*	OEColor
stub column font	*GetStubColumnFont*	*SetStubColumnFont*	*OEFont*

See also

- *Depicting Molecules in a Table* section
- *OEImageTable* class

Constructors

```
OEImageTableOptions(unsigned int rows, unsigned int cols,
                    unsigned int style = OEImageTableStyle::Default)
```

Creates an *OEImageTableOptions* object with the given parameters.

rows
: The total number of rows in the table including the header.

cols
The total number of columns in the table including the stub column.

style
This value has to be from the *OEImageTableStyle* namespace.

Note

By default, the columns and rows of a table are created with equal widths and heights. The relative width and height ratio of the columns and rows can be adjusted using the *OEImageTableOptions.SetColumnWidths* and *OEImageTableOptions.SetRowHeights* methods, respectively.

Example (*Example of generating a 4 x 4 image table with header and 'MediumBlue' style*)

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
```

(header 1)	(header 2)	(header 3)	(header 4)
(body 1, 1)	(body 1, 2)	(body 1, 3)	(body 1, 4)
(body 2, 1)	(body 2, 2)	(body 2, 3)	(body 2, 4)
(body 3, 1)	(body 3, 2)	(body 3, 3)	(body 3, 4)

Fig. 187: **Example of generating a 4 x 4 image table with header and 'MediumBlue' style**

See also

- *OEImageTableStyle* namespace
- *OEImageTableOptions.NumRows* method
- *OEImageTableOptions.NumColumns* method

```
OEImageTableOptions(const OEImageTableOptions& rhs)
```

Copy constructor.

GetCellBorderPen

```
const OEPen& GetCellBorderPen() const
```

Returns the pen that is used to mark the border of the cells in the table.

See also

- *OEPen* class
- *OEImageTableOptions.SetCellBorderPen* method

GetCellColor

```
const OESystem::OEColor GetCellColor(bool even = true) const
```

See also

- *OEImageTableOptions.SetCellColor* method

Returns the color used to highlight the 'body' cells in the table. A cell is considered to be part of the 'body' of the table if it is neither in the header nor in the stub column.

even
: This parameter determines whether to return the color of the even or odd rows in the table.

See also

- OEColor class
- *OEImageTableOptions.SetCellColor* method

GetCellFont

```
const OEFont GetCellFont() const
```

Returns the font used by the *OEImageTable.DrawText* method to draw text on the 'body' cells.

See also

- *OEFont* class
- *OEImageTableOptions.SetCellFont* method

GetHeaderColor

```
const OESystem::OEColor GetHeaderColor() const
```

Returns the color used to highlight the 'header' cells in the table.

See also

- OEColor class
- *OEImageTableOptions.SetHeaderColor* method

GetHeaderFont

```
const OEFont GetHeaderFont() const
```

Returns the font used by the *OEImageTable.DrawText* method to draw text on the 'header' cells.

See also

- *OEFont* class
- *OEImageTableOptions.SetHeaderFont* method

GetMargin

```
double GetMargin(unsigned int margin) const
```

Returns the ratio of a specific margin of the table.

margin
This value has to be from the *OEMargin* namespace.

See also

- *OEImageTableOptions.SetMargin* method
- *OEImageTableOptions.SetMargins* method

GetStubColumnColor

```
const OESystem::OEColor GetStubColumnColor() const
```

Returns the color used to highlight the 'stub column' cells in the table.

See also

- OEColor class
- *OEImageTableOptions.SetStubColumnColor* method

GetStubColumnFont

```
const OEFont GetStubColumnFont() const
```

Returns the font used by the *OEImageTable.DrawText* method to draw text on the 'stub column' cells.

See also

- *OEFont* class
- *OEImageTableOptions.SetStubColumnFont* method

HasHeader

```
bool HasHeader() const
```

Returns whether a table has a 'header' row. By default, a table is generated with a 'header' row.

See also

- *OEImageTableOptions.SetHeader* method
- *OEImageTableOptions.HasStubColumn* method
- *OEImageTableOptions.SetStubColumn* method

HasStubColumn

```
bool HasStubColumn() const
```

Returns whether the table has a 'stub' column. By default, a table is generated without a 'stub' column.

See also

- *OEImageTableOptions.SetStubColumn* method
- *OEImageTableOptions.HasHeader* method
- *OEImageTableOptions.SetHeader* method

NumColumns

```
unsigned int NumColumns() const
```

Returns the total number of columns in the table including the stub column.

NumRows

```
unsigned int NumRows() const
```

Returns the total number of rows in the table including the header row.

SetBaseFontSize

```
void SetBaseFontSize(unsigned int fontsize)
```

Controls the font size in add table cells (including cells of header, stub column and body).

See also

The other properties of the used fonts can be controlled with the following methods:

- *OEImageTableOptions.SetCellFont* method
- *OEImageTableOptions.SetHeaderFont* method
- *OEImageTableOptions.SetStubColumnFont* method

SetCellBorderPen

```
void SetCellBorderPen(const OEPen& pen)
```

Sets the pen that is used to mark the border of the cells in the table.

Example (*Example of using the SetCellBorderPen method*)

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
OEPen pen = new OEPen(oechem.getOEBlack(), oechem.getOEDarkBlue(), OEFill.Off, 2.0);
tableopts.SetCellBorderPen(pen);
```

(header 1)	(header 2)	(header 3)	(header 4)
(body 1, 1)	(body 1, 2)	(body 1, 3)	(body 1, 4)
(body 2, 1)	(body 2, 2)	(body 2, 3)	(body 2, 4)
(body 3, 1)	(body 3, 2)	(body 3, 3)	(body 3, 4)

Fig. 188: **Example of using the SetCellBorderPen method**

See also

- *OEPen* class
- *OEImageTableOptions.GetCellBorderPen* method

SetCellColor

```
void SetCellColor(const OESystem::OEColor& color, bool even = true)
```

Sets the color used to highlight the 'body' cells in the table. A cell is considered to be part of the 'body' of the table if it is neither in the header nor in the stub column.

color
The color of the cells.

even
This parameter determines whether the color is applied to the even or to the odd rows of the table.

Example (*Example of using the SetCellColor method*)

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
boolean even = true;
tableopts.SetCellColor(oechem.getOELightGrey(), !even);
```

(header 1)	(header 2)	(header 3)	(header 4)
(body 1, 1)	(body 1, 2)	(body 1, 3)	(body 1, 4)
(body 2, 1)	(body 2, 2)	(body 2, 3)	(body 2, 4)
(body 3, 1)	(body 3, 2)	(body 3, 3)	(body 3, 4)

Fig. 189: **Example of using the SetCellColor method**

See also

- OEColor class
- *OEImageTableOptions.GetCellColor* method

SetCellFont

```
void SetCellFont(const OEFont& font)
```

Sets the font used by the *OEImageTable.DrawText* method to draw text on the 'body' cells. A cell is considered to be part of the 'body' of the table if it is neither in the header nor in the stub column.

Example (*Example of using the SetCellFont method*)

(header 1)	(header 2)	(header 3)	(header 4)
(body 1, 1)	*(body 1, 2)*	*(body 1, 3)*	*(body 1, 4)*
(body 2, 1)	*(body 2, 2)*	*(body 2, 3)*	*(body 2, 4)*
(body 3, 1)	*(body 3, 2)*	*(body 3, 3)*	*(body 3, 4)*

Fig. 190: **Example of using the SetCellFont method**

See also

- *OEImageTableOptions.GetCellFont* method

SetColumnWidths

```
bool SetColumnWidths(const std::vector<unsigned int>& widths)
```

Sets the relative widths of the columns.

widths
The relative widths of the columns. The size of the vector has to be *OEImageTableOptions.NumColumns()*.

Example (*Example of using the SetColumnWidths method*)

The following example show how to adjust the width ratio of the columns.

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
ArrayList<Integer> widths = new ArrayList<Integer>();
widths.add(20);
widths.add(10);
widths.add(20);
widths.add(10);
tableopts.SetColumnWidths(widths);
```

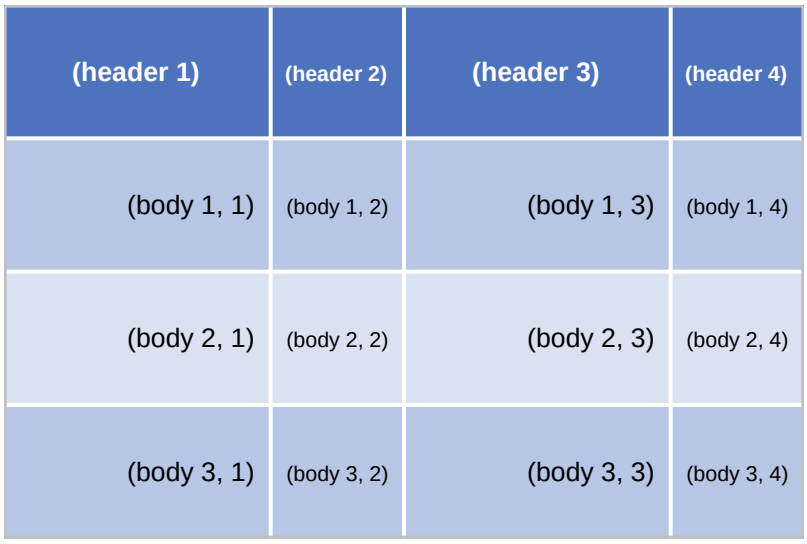


Fig. 191: **Example of using the SetColumnWidths method**

See also

- *OEImageTableOptions.SetRowHeights* method

SetHeader

```
bool SetHeader(bool has)
```

Sets whether to have a 'header' row in the table.

Example (*Example of using the SetHeader method*)

By default, a table is generated with a header row on the top. The following example show how to remove the header.

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
tableopts.SetHeader(false);
```

(body 1, 1)	(body 1, 2)	(body 1, 3)	(body 1, 4)
(body 2, 1)	(body 2, 2)	(body 2, 3)	(body 2, 4)
(body 3, 1)	(body 3, 2)	(body 3, 3)	(body 3, 4)
(body 4, 1)	(body 4, 2)	(body 4, 3)	(body 4, 4)

Fig. 192: **Example of using the SetHeader method**

See also

- *OEImageTableOptions.HasHeader* method
- *OEImageTableOptions.HasStubColumn* method
- *OEImageTableOptions.SetStubColumn* method

SetHeaderColor

```
void SetHeaderColor(const OESystem::OEColor& color)
```

Sets the color used to highlight the 'header' cells in the table.

Example (*Example of using the SetHeaderColor method*)

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
tableopts.SetHeaderColor(oechem.getOELightGrey());
```

(header 1)	(header 2)	(header 3)	(header 4)
(body 1, 1)	(body 1, 2)	(body 1, 3)	(body 1, 4)
(body 2, 1)	(body 2, 2)	(body 2, 3)	(body 2, 4)
(body 3, 1)	(body 3, 2)	(body 3, 3)	(body 3, 4)

Fig. 193: **Example of using the SetHeaderColor method**

See also

- OEColor class
- *OEImageTableOptions.GetHeaderColor* method

SetHeaderFont

```
void SetHeaderFont(const OEFont& font)
```

Sets the font used by the *OEImageTable.DrawText* method to draw text on the 'header' cells.

Example (*Example of using the SetHeaderFont method*)

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
OEFont font = new OEFont(OEFontFamily.Default, OEFontStyle.Default,
                         12, OEAlignment.Right, oechem.getOEPinkTint());
tableopts.SetHeaderFont(font);
```

(header 1)	(header 2)	(header 3)	(header 4)
(body 1, 1)	(body 1, 2)	(body 1, 3)	(body 1, 4)
(body 2, 1)	(body 2, 2)	(body 2, 3)	(body 2, 4)
(body 3, 1)	(body 3, 2)	(body 3, 3)	(body 3, 4)

Fig. 194: **Example of using the SetHeaderFont method**

See also

- *OEFont* class
- `OEImageTableOptions.GetHeaderFont` method

SetMargin

```
bool SetMargin(unsigned int marginloc, double margin)
```

Sets the size of a specific margin of the table.

marginloc
This value has to be from the `OEMargin` namespace.

margin
This number is considered as a percentage of either the width or the height of image on which the table will be drawn and has to be in the range of $[0.0, 25.0]$. For example, 10.0% means, that the left and right margin are 10% of the total width of the image, and the top and bottom margins are 10% of the total height of the image.

Example (*Example of using the SetMargin method*)

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
tableopts.SetMargin(OEMargin.Left, 10.0);
tableopts.SetMargin(OEMargin.Right, 10.0);
```



Fig. 195: **Example of using the SetMargin method**

See also

- *OEImageTableOptions.GetMargin* method
- *OEImageTableOptions.SetMargins* method

SetMargins

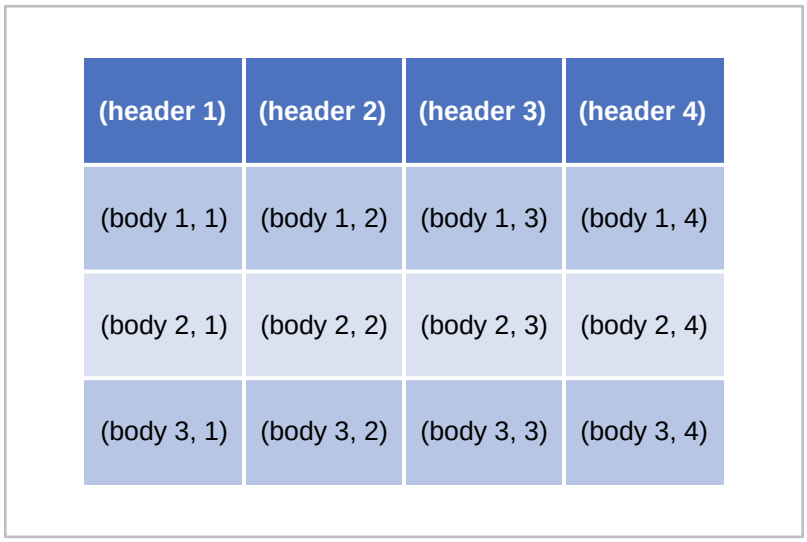
```
bool SetMargins(double margin)
```

Sets the size of all margins of the table.

margin
: This number is considered as a percentage of either the width or the height of image on which the table will be drawn and has to be in the range of $[0.0, 25.0]$. For example, 10.0% means, that the left and right margin are 10% of the total width of the image, and the top and bottom margins are 10% of the total height of the image.

Example (*Example of using the SetMargins method*)

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
tableopts.SetMargins(10.0);
```



(header 1)	(header 2)	(header 3)	(header 4)
(body 1, 1)	(body 1, 2)	(body 1, 3)	(body 1, 4)
(body 2, 1)	(body 2, 2)	(body 2, 3)	(body 2, 4)
(body 3, 1)	(body 3, 2)	(body 3, 3)	(body 3, 4)

Fig. 196: **Example of using the SetMargins method**

See also

- *OEImageTableOptions.GetMargin* method
- *OEImageTableOptions.SetMargin* method

SetRowHeights

```
bool SetRowHeights(const std::vector<unsigned int>& heights)
```

Sets the relative heights of the rows.

heights
The relative heights of the rows. The size of the vector has to be *OEImageTableOptions.NumRows()*.

Example (*Example of using the SetRowHeights method*)

The following example show how to adjust the height ratio of the rows.

```
EImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
rrayList<Integer> heights = new ArrayList<Integer>();
eights.add(10);
eights.add(20);
eights.add(30);
eights.add(40);
ableopts.SetRowHeights(heights);
```

(header 1)	(header 2)	(header 3)	(header 4)
(body 1, 1)	(body 1, 2)	(body 1, 3)	(body 1, 4)
(body 2, 1)	(body 2, 2)	(body 2, 3)	(body 2, 4)
(body 3, 1)	(body 3, 2)	(body 3, 3)	(body 3, 4)

Fig. 197: **Example of using the SetRowHeights method**

See also

- *OEImageTableOptions.SetColumnWidths* method

SetStubColumn

```
bool SetStubColumn(bool has)
```

Sets whether to have a 'stub' column in the table.

Example (*Example of using the SetStubColumn method*)

By default, a table is generated without a 'stub' column. The following example show how to add one on the left.

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
tableopts.SetStubColumn(true);
```

	(header 1)	(header 2)	(header 3)
(stub 1)	(body 1, 1)	(body 1, 2)	(body 1, 3)
(stub 2)	(body 2, 1)	(body 2, 2)	(body 2, 3)
(stub 3)	(body 3, 1)	(body 3, 2)	(body 3, 3)

Fig. 198: **Example of using the SetStubColumn method**

See also

- *OEImageTableOptions.HasStubColumn* method
- *OEImageTableOptions.HasHeader* method
- *OEImageTableOptions.SetHeader* method

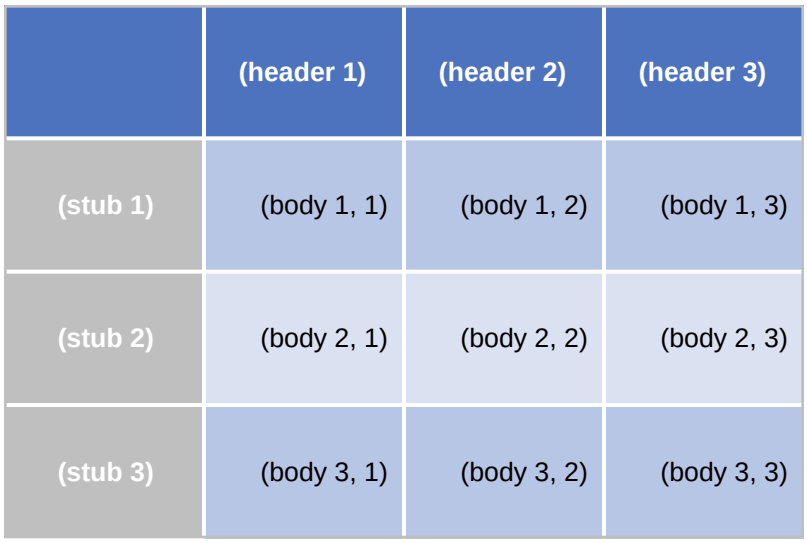
SetStubColumnColor

```
void SetStubColumnColor(const OESystem::OEColor & color)
```

Sets the color used to highlight the 'stub' cells in the table.

Example (*Example of using the SetStubColumnColor method*)

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
tableopts.SetStubColumn(true);
tableopts.SetStubColumnColor(oechem.getOELightGrey());
```



	(header 1)	(header 2)	(header 3)
(stub 1)	(body 1, 1)	(body 1, 2)	(body 1, 3)
(stub 2)	(body 2, 1)	(body 2, 2)	(body 2, 3)
(stub 3)	(body 3, 1)	(body 3, 2)	(body 3, 3)

Fig. 199: **Example of using the SetStubColumnColor method**

See also

- OEColor class
- *OEImageTableOptions.GetStubColumnColor* method

SetStubColumnFont

```
void SetStubColumnFont(const OEFont& font)
```

Sets the font used by the *OEImageTable.DrawText* method to draw text on the 'stub column' cells.

Example (*Example of using the SetStubColumnFont method*)

```
OEImageTableOptions tableopts = new OEImageTableOptions(4, 4, OEImageTableStyle.
↪MediumBlue);
tableopts.SetStubColumn(true);
OEFont font = new OEFont(OEFontFamily.Default, OEFontStyle.Default,
                         12, OEAlignment.Right, oechem.getOEPinkTint());
tableopts.SetStubColumnFont(font);
```

	(header 1)	(header 2)	(header 3)
(stub 1)	(body 1, 1)	(body 1, 2)	(body 1, 3)
(stub 2)	(body 2, 1)	(body 2, 2)	(body 2, 3)
(stub 3)	(body 3, 1)	(body 3, 2)	(body 3, 3)

Fig. 200: **Example of using the SetStubColumnFont method**

See also

- *OEFont* class
- *OEImageTableOptions.GetStubColumnFont* method

4.1.44 OEJavaImageBase

```
class OEJavaImageBase extends OEImageBase
```

The *OEJavaImageBase* is an an implementation of *OEImageBase* that uses a java.awt.Graphics object as a canvas to draw on. This may be faster in some circumstances, since the JNI layer is avoided.

The following Java example demonstrates the usage of *OEJavaImageBase* class.

Download code

JavaImageBase.java

The following methods are publicly inherited from *OEImageBase*:

Clear	*DrawPie*	*DrawTriangle*
DrawArc	*DrawPoint*	*GetGlobalOffset*
DrawCircle	*DrawPolygon*	*GetHeight*
DrawCubicBezier	*DrawQuadraticBezier*	*GetMinFontSize*
DrawLine	*DrawRectangle*	*GetWidth*
DrawPath	*DrawText*	*GetSVGClass*
GetSVGGroup	*NewSVGClass*	*NewSVGGroup*
PopGroup	*PushGroup*	*SetMinFontSize*

Constructors

```
OEJavaImageBase(java.awt.Graphics graphics, double width,
                double height)
```

Default constructor that creates an *OEJavaImageBase* with the specified width and height.

graphics
A java.awt.Graphics object.

width, height
The dimensions of the image, both have to be positive (non-zero) numbers.

SetAntiAliasing

```
void SetAntiAliasing(boolean state)
```

Enables or disables antialiasing on the Graphics object. Antialiasing is on by default.

GetAntiAliasing

```
boolean GetAntiAliasing()
```

Returns a boolean indicating if antialiasing is on.

GetGraphics

```
Graphics GetGraphics()
```

Returns the graphics object used to construct this object with.

4.1.45 OELegendLayout

```
class OELegendLayout
```

This class represents *OELegendLayout* which is a layout manager that allows to create interactive legends in SVG image format.

See also

- *OELegendLayoutOptions* class
- *OEDrawLegendLayout* function
- *OELegendLayoutStyle* namespace
- *OELegendColorStyle* namespace
- *OELegendInteractiveStyle* namespace

Constructors

```
OELegendLayout(OEImageBase& parent, const std::string& text,
               const OELegendLayoutOptions& opts)
```

Creates an *OELegendLayout* object with the specified options.

parent
: The *OEImageBase* object on which the *OELegendLayout* object is placed.

text
: The text of the button of the legend.

opts
: The *OELegendLayoutOptions* object that defines the properties and style of the legend layout.

See also

- `OELegendLayout.GetOptions` method

GetLegendArea

```
OEImage& GetLegendArea()
```

Provides access to the legend area of the *OELegendLayout* object.

See also

- *OEImage* class

GetOptions

```
const OELegendLayoutOptions& GetOptions() const
```

Returns the options used to initialized the *OELegendLayout* object.

See also

- *OELegendLayoutOptions* class

4.1.46 OELegendLayoutOptions

```
class OELegendLayoutOptions
```

This class represents the *OELegendLayoutOptions* class that encapsulates properties that determine how an legend layout is depicted.

The *OELegendLayoutOptions* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
legend layout style	*GetLayoutStyle*	(set by the constructor)	*OELegendLayoutStyle* namespace
legend color style	*GetColorStyle*	(set by the constructor)	*OELegendColorStyle* namespace
legend interactive style	*GetInteractiveStyle*	(set by the constructor)	*OELegendInteractiveStyle* namespace
button label font	*GetButtonFont*	*SetButtonFont*	*OEFont*
button pen(s)	*GetButtonPen*	*SetButtonPen*	*OEPen*
button height scale	*GetButtonHeightScal*	*SetButtonHeightScal*	positive floating point number in the range of $[0.5, 3.0]$
button width scale	*GetButtonWidthScale*	*SetButtonWidthScale*	positive floating point number in the range of $[0.5, 5.0]$
legend box pen	*GetLegendBoxPen*	*SetLegendBoxPen*	*OEPen*
layout margins	*GetMargin*	*SetMargin* / *SetMargins*	positive floating point number in the range of $[0.0, 90.0]$

See also

- *OELegendLayout* class

Constructors

```
OELegendLayoutOptions(unsigned int layout = OELegendLayoutStyle::Default,
                      unsigned int color = OELegendColorStyle::Default,
                      unsigned int inter = OELegendInteractiveStyle::Default)
```

See also

- *OELegendLayoutStyle* namespace
- *OELegendColorStyle* namespace
- *OELegendInteractiveStyle* namespace

```
OELegendLayoutOptions(const OELegendLayoutOptions& rhs)
```

Copy constructor.

GetButtonFont

```
const OEFont& GetButtonFont() const
```

See also

- *OEFont* class
- *OELegendLayoutOptions.SetButtonFont* method

GetButtonHeightScale

```
double GetButtonHeightScale() const
```

See also

- *OELegendLayoutOptions.SetButtonHeightScale* method

GetButtonPen

```
const OEPen& GetButtonPen(bool on) const
```

See also

- *OEPen* class
- *OELegendLayoutOptions.SetButtonPen* method

GetButtonWidthScale

```
double GetButtonWidthScale() const
```

See also

- *OELegendLayoutOptions.SetButtonWidthScale* method

GetColorStyle

```
unsigned int GetColorStyle() const
```

See also

- *OELegendColorStyle* namespace

GetInteractiveStyle

```
unsigned int GetInteractiveStyle() const
```

See also

- *OELegendInteractiveStyle* namespace

GetLayoutStyle

```
unsigned int GetLayoutStyle() const
```

See also

- *OELegendLayoutStyle* namespace

GetLegendBoxPen

```
const OEPen& GetLegendBoxPen() const
```

See also

- *OEPen* class
- *OELegendLayoutOptions.SetLegendBoxPen* method

GetMargin

```
double GetMargin(unsigned int percent) const
```

Returns the ratio of a specific margin of the legend layout.

margin
: This value has to be from the *OEMargin* namespace.

See also

- *OELegendLayoutOptions.SetMargin* method
- *OELegendLayoutOptions.SetMargins* method

SetButtonFont

```
void SetButtonFont(const OEFont& font)
```

See also

- *OEFont* class
- *OELegendLayoutOptions.GetButtonFont* method

SetButtonHeightScale

```
void SetButtonHeightScale(double scale)
```

See also

- *OELegendLayoutOptions.GetButtonHeightScale* method

SetButtonPen

```
void SetButtonPen(const OEPen& pen, bool on)
```

See also

- *OEPen* class
- *OELegendLayoutOptions.GetButtonPen* method

SetButtonWidthScale

```
void SetButtonWidthScale(double s)
```

See also

- *OELegendLayoutOptions.GetButtonWidthScale* method

SetLegendBoxPen

```
void SetLegendBoxPen(const OEPen& pen)
```

See also

- *OEPen* class
- *OELegendLayoutOptions.GetLegendBoxPen* method

SetMargin

```
bool SetMargin(unsigned int margin, double percent)
```

Sets the size of a specific margin of the legend layout.

marginloc
: This value has to be from the *OEMargin* namespace.

margin
: This number is considered as a percentage of either the width or the height of image on which the legend layout will be drawn and has to be in the range of $[0.0, 90.0]$. For example, 10.0% means, that the left and right margin are 10% of the total width of the image, and the top and bottom margins are 10% of the total height of the image.

See also

- *OELegendLayoutOptions.GetMargin* method
- *OELegendLayoutOptions.SetMargins* method

SetMargins

```
bool SetMargins(double margin)
```

Sets the size of all margins of the legend layout.

margin
: This number is considered as a percentage of either the width or the height of image on which the legend layout will be drawn and has to be in the range of $[0.0, 90.0]$. For example, 10.0% means, that the left and right margin are 10% of the total width of the image, and the top and bottom margins are 10% of the total height of the image.

See also

- *OELegendLayoutOptions.GetMargin* method
- *OELegendLayoutOptions.SetMargin* method

4.1.47 OEMolDisplayBase

```
class OEMolDisplayBase : public OESystem::OEBase
```

The *OEMolDisplayBase* class is the abstract interface for representing molecule display information within **OEDepict TK**. See *Figure: OEDepict TK molecule display class hierarchy*.

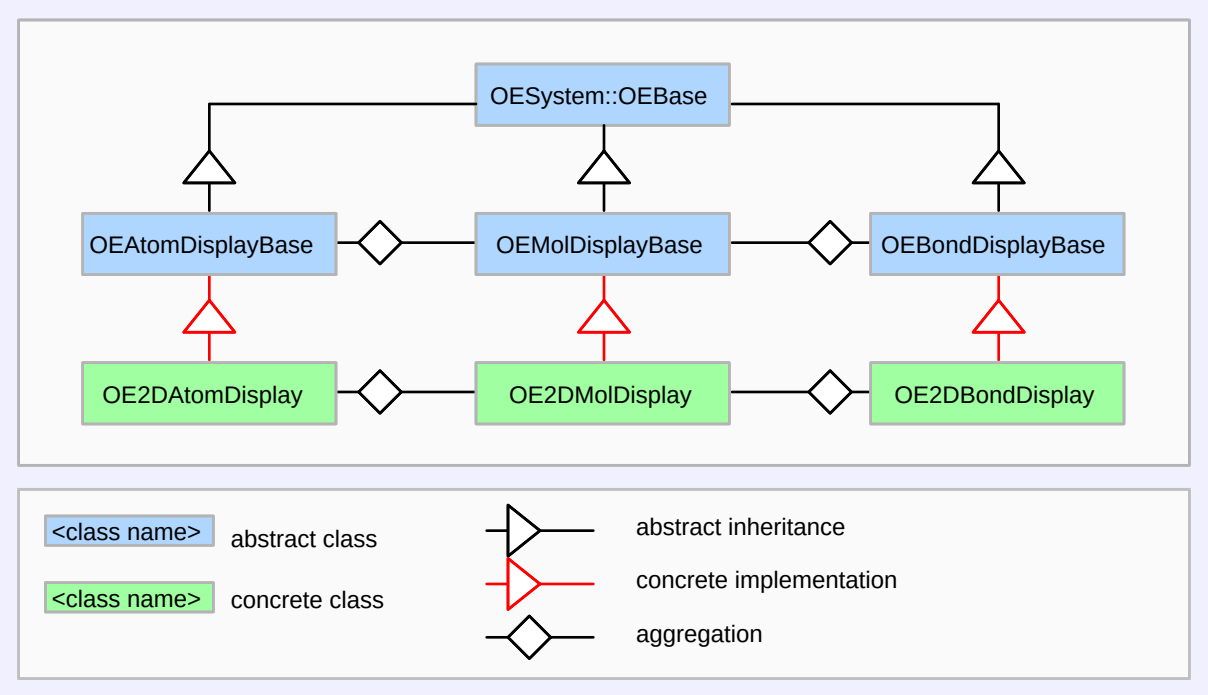


Fig. 201: **OEDepict TK molecule display class hierarchy**

See also

- *OEAtomDisplayBase* class
- *OEBondDisplayBase* class

The following classes derive from this class:

- *OE2DMolDisplay*

GetDataType

```
const void* GetDataType() const
```

This function is used to perform run-time type identification.

See also

- `OEBase.GetDataType` method in the **OEChem TK** manual

GetMolecule

```
const OEChem::OEMolBase* GetMolecule() const
```

Returns the const pointer of the OEMolBase object for which the display properties are stored in the class derived from the *OEMolDisplayBase* abstract class.

IsDataType

```
bool IsDataType(const void *) const
```

Returns whether type is the same as the instance this method is called on.

See also

- `OEBase.IsDataType` method in the **OEChem TK** manual

4.1.48 OEMultiPageImageFile

```
class OEMultiPageImageFile
```

The *OEMultiPageImageFile* class supports the creation of multi-page images. The dimensions and orientation of the pages are determined when an *OEMultiPageImageFile* object is created. Pages are created by calling the `OEMultiPageImageFile.NewPage` method. See *Figure: Schematic representation of an OEMultiPageImageFile object*.

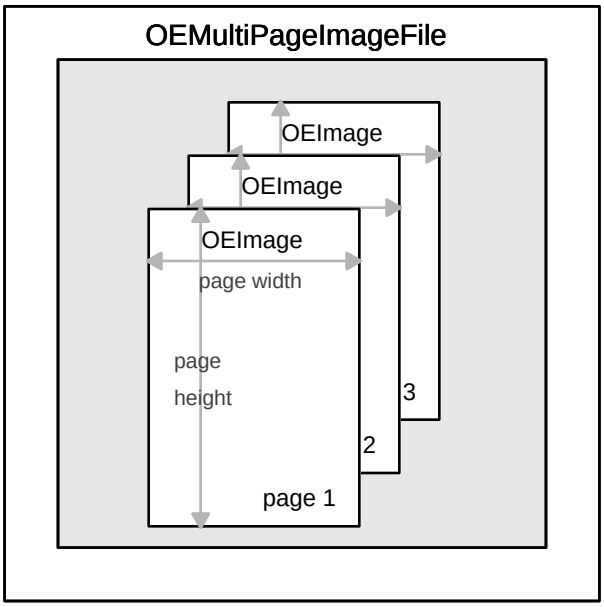


Fig. 202: **Schematic representation of an OEMultiPageImageFile object**

See also

- *Multi Page Images* chapter
- `OEGetSupportedMultiPageImageFileExtensions` function

Hint

We highly recommend to usage of the more convenient high-level *OEReport* class when generating multi-page images.

- *Multi Page Reports* section
- *Depicting Molecules in Multi-Page* example
- *Depicting MDL Query Substructure Search Hits* example

Constructors

```
OEMultiPageImageFile(double pagewidth, double pageheight)
```

Creates an *OEMultiPageImageFile* object with a specified page width and height. Both the width and the height parameters have to be positive (non-zero) numbers.

```
OEMultiPageImageFile(unsigned int orientation,
                     unsigned int imagesize)
```

Creates an *OEMultiPageImageFile* object with the given orientation and size.

orientation
Determine the orientation of the pages. This value has to be from the `OEPageOrientation` namespace.

imagesize
Determine the width and height of the pages. This value has to be from the *OEPageSize* namespace.

See also

- *OEGetPageHeight* function
- *OEGetPageWidth* function

GetOrientation

```
unsigned int GetOrientation() const
```

Returns the orientation of the pages. The return value is taken from the *OEPageOrientation* namespace.

GetPage

```
OEImage* GetPage(unsigned int page)
const OEImage* GetPage(unsigned int page) const
```

Returns the p^{th} page .

p
This value has to be in the range from *1* to *OEMultiPageImageFile.NumPages()*.

Note

If p page index is out of range, then the *OEMultiPageImageFile.GetPage* method returns a NULL pointer.

GetPageHeight

```
double GetPageHeight() const
```

Returns the height of the pages.

GetPageWidth

```
double GetPageWidth() const
```

Returns the width of the pages.

GetPages

```
OESystem::OEIterBase<OEImage>* GetPages()
OESystem::OEIterBase<const OEImage>* GetPages() const
```

Returns an iterator over all pages stored in the *OEMultiPageImageFile* object.

NewPage

```
OEImage& NewPage()
```

Returns a new page as an *OEImage* object.

NumPages

```
unsigned int NumPages() const
```

Returns the number of pages stored in the *OEMultiPageImageFile* object.

4.1.49 OENearestAtom

```
class OENearestAtom
```

The *OENearestAtom* class that is used by *OEGetNearestAtom* and *OEGetNearbyAtom* functions. It stores an atom pointer, an associated distance and the display position of the atom.

Constructors

```
OENearestAtom(const OEChem::OEAtomBase* atom, double distance)
OENearestAtom(const OEChem::OEAtomBase* atom, double distance,
              const OE2DPoint& displaypos)
```

Constructors.

If no display position is given then it will be initialized to *OE2DPoint(0.0, 0.0)*.

GetAtom

```
const OEChem::OEAtomBase* GetAtom() const
```

Returns the atom pointer stored in the *OENearestAtom* object.

GetDisplayPosition

```
const OE2DPoint& GetDisplayPosition() const
```

Returns the display position of the nearest atom.

GetDist

```
double GetDist() const
```

Returns the distance associated with the atom.

IsValid

```
bool IsValid() const
```

Returns whether the *OENearestAtom* object is initialized correctly.

4.1.50 OENearestBond

```
class OENearestBond
```

The *OENearestBond* class, used by *OEGetNearestBond* and *OEGetNearbyBond* functions. It stores a bond pointer, the associated distance and the display position of the bond.

Constructors

```
OENearestBond()
OENearestBond(const OEChem::OEBondBase* bond, double dist)
OENearestBond(const OEChem::OEBondBase* bond, double dist,
              const OE2DPoint& pos)
OENearestBond(const OENearestBond&)
```

Constructors.

If no display position is given then it will be initialized to *OE2DPoint(0.0, 0.0)*.

GetBond

```
const OEChem::OEBondBase* GetBond() const
```

Returns the bond pointer stored in the *OENearestBond* object.

GetDisplayPosition

```
const OE2DPoint& GetDisplayPosition() const
```

Returns the display position of the middle of the nearest bond.

GetDist

```
double GetDist() const
```

Returns the distance associated with the bond.

IsValid

```
bool IsValid() const
```

Returns whether the *OENearestBond* object is initialized correctly.

4.1.51 OEPen

```
class OEPen
```

The *OEPen* class encapsulates properties that determine the way various lines, curves and shapes are drawn.

Currently the *OEPen* class stores the following properties:

Property	Get method	Set method	Corresponding namespace/class
background color	*OEPen.GetBackColor*	*OEPen.SetBackColor*	OEColor
foreground color	*OEPen.GetForeColor*	*OEPen.SetForeColor*	OEColor
fill	*OEPen.GetFill*	*OEPen.SetFill*	*OEFill*
line join	*OEPen.GetLineJoin*	*OEPen.SetLineJoin*	*OELineJoin*
line cap	*OEPen.GetLineCap*	*OEPen.SetLineCap*	*OELineCap*
line width	*OEPen.GetLineWidth*	*OEPen.SetLineWidth*	none
line stipple	*OEPen.SetLineStipple*	*OEPen.GetLineStipple*	*OEStipple*
stipple factor	*OEPen.SetStippleFactor*	*OEPen.GetStippleFactor*	none

Constructors

```
OEPen()
```

Default constructor that initializes an *OEPen* object with the following properties:

Table 10: Default parameters of OEPen

Property	Default value
background color	`OEWhite` = OEColor(255,255,255)
foreground color	`OEBlack` = OEColor(0,0,0)
fill	*OEFill.Default*
line join	*OELineJoin.Default*
line cap	*OELineCap.Default*
line width	1.0
line stipple	*OEStipple.Default*
stipple factor	1

```
OEPen(const OESystem::OEColor& bg, const OESystem::OEColor& fg,
      unsigned int fill = OEFill::Off, double linewidth = 1.0,
      unsigned short stipple = OEStipple::Solid)
```

Creates and initializes an *OEPen* object with the given parameters. The other properties of the pen are set to the default values (see table above).

bg, fg
: The background and foreground color of the pen.

fill
: The *fill* property of the pen.

linewidth
: The *line width* property of the pen.

```
OEPen(const OEPen& rhs)
```

Copy constructor.

GetBackColor

```
const OESystem::OEColor& GetBackColor() const
```

Returns the background color of the *OEPen* object that is used to fill shapes.

See also

- *OEPen.SetBackColor* method
- *OEPen.SetFill* method
- OEColor class

GetFill

```
unsigned int GetFill() const
```

If the returned value is *OEFill.On*, than shapes drawn with the *OEPen* object are filled with the background color of the pen. If the returned value is *OEFill.Off*, than shapes drawn with the *OEPen* object are left "empty" *i.e.* only their outlines are drawn using the foreground color of the pen.

See also

- *OEPen.SetFill* method
- *OEPen.SetBackColor* method
- *OEFill* namespace

GetForeColor

```
const OESystem::OEColor& GetForeColor() const
```

Returns the foreground color of the *OEPen* object that controls the color of the strokes when drawing lines, curves and outlines of shapes.

See also

- *OEPen.SetForeColor* method
- OEColor class

GetLineCap

```
unsigned int GetLineCap() const
```

Returns the *line cap* property of the *OEPen* object that controls how the endpoints of lines are drawn. The return value is taken from the *OELineCap* namespace.

See also

- *OEPen.SetLineCap* method
- *OELineCap* namespace

GetLineJoin

```
unsigned int GetLineJoin() const
```

Returns the *line join* property of the *OEPen* object that controls how lines connect at corners when drawing outlines. The return value is taken from the *OELineJoin* namespace.

See also

- *OEPen.SetLineJoin* method
- *OELineJoin* namespace

GetLineStipple

```
unsigned short GetLineStipple() const
```

Returns the *line stipple* property of the *OEPen* object that controls the pattern of dashes and gaps used to draw lines and outlines. The return value is taken from the *OEStipple* namespace.

See also

- *OEPen.SetLineStipple* method
- *OEPen.SetStippleFactor* method
- *OEStipple* namespace

GetLineWidth

```
double GetLineWidth() const
```

Returns the *line width* of the *OEPen* object that controls the width of strokes used when drawing lines, curves and outlines of shapes.

See also

- *OEPen.SetLineWidth* method

GetStippleFactor

```
unsigned short GetStippleFactor() const
```

Returns the *stipple factor* of the *OEPen* object. The stipple factor is a multiplier that is used when drawing stipple patterns. For example, if the factor is three, then the stipple pattern is stretched to three times of its original size.

See also

- *OEPen.SetStippleFactor* method
- *OEPen.SetLineStipple* method

SetBackColor

```
void SetBackColor(const OESystem::OEColor& col)
```

Sets the background color of the *OEPen* object that is used to fill shapes.

Example (*Figure: Example of using the SetBackColor method*)

```
OEPen pen = new OEPen();
pen.SetFill(OEFill.On);
pen.SetBackColor(oechem.getOEBlueTint());
```

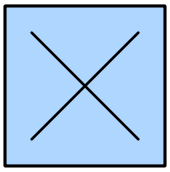


Fig. 203: **Example of using the SetBackColor method**

Note

In order to fill a shape using the background color, the fill property of the *OEPen* has to be turned on (*OEFill.On*). (See *OEPen.SetFill* method and *OEFill* namespace)

See also

- *OEPen.GetBackColor* method
- *OEPen.GetFill* method
- OEColor class

SetFill

```
void SetFill(unsigned int fill)
```

Sets the *fill* property of the *OEPen* object.

fill
: This value has to be from the *OEFill* namespace.

See also

- *OEPen.GetFill* method
- *OEPen.GetBackColor* method
- *OEFill* namespace

SetForeColor

```
void SetForeColor(const OESystem::OEColor& col)
```

Sets the foreground color of the *OEPen* object that controls the color of the stroke when drawing lines, curves and outlines of shapes.

Example (*Figure: Example of using the SetForeColor method*)

```
OEPen pen = new OEPen();
pen.SetForeColor(oechem.getOERed());
```



Fig. 204: **Example of using the SetForeColor method**

See also

- *OEPen.GetForeColor* method
- OEColor class

SetLineCap

```
void SetLineCap(unsigned int cap)
```

Sets the *line cap* property of the *OEPen* object that controls how the endpoints of lines are drawn.

cap
: This value has to be from the *OELineCap* namespace.

Example (*Figure: Example of using the SetLineCap method*)

```
OEPen pen = new OEPen();
pen.SetLineWidth(4);
pen.SetLineCap(OELineCap.Square);
```



Fig. 205: **Example of using the SetLineCap method**

See also

- *OEPen.GetLineCap* method
- *OELineCap* namespace

SetLineJoin

```
void SetLineJoin(unsigned int join)
```

Sets the *line join* property of the *OEPen* object that controls how lines connect at corners when drawing outlines.

join
: This value has to be from the *OELineJoin* namespace.

Example (*Figure: Example of using the SetLineJoin method*)

```
OEPen pen = new OEPen();
pen.SetLineWidth(4);
pen.SetLineJoin(OELineJoin.Miter);
```



Fig. 206: **Example of using the SetLineJoin method**

See also

- *OEPen.GetLineJoin* method
- *OELineJoin* namespace

SetLineStipple

```
void SetLineStipple(unsigned short pattern)
```

Sets the *line stipple* property of the *OEPen* object that controls the pattern of dashes and gaps used to draw lines and outlines.

pattern
: This value has to be from the *OEStipple* namespace.

Example (*Figure: Example of using the SetLineStipple method*)

```
OEPen pen = new OEPen();
pen.SetLineStipple(OEStipple.ShortDash);
```



Fig. 207: **Example of using the SetLineStipple method**

See also

- *OEPen.SetLineStipple* method
- *OEPen.SetStippleFactor* method
- *OEStipple* namespace

SetLineWidth

```
void SetLineWidth(double width)
```

Sets the *line width* of the *OEPen* object that controls the width of strokes used when drawing lines, curves and outlines of shapes.

width
: This value has to be a positive number.

Example (*Figure: Example of using the SetLineWidth method*)

```
OEPen pen = new OEPen();
pen.SetLineWidth(5.0);
```



Fig. 208: **Example of using the SetLineWidth method**

See also

- *OEPen.GetLineWidth* method

SetStippleFactor

```
void SetStippleFactor(unsigned short factor)
```

Sets the *stipple factor* of the *OEPen* object. The stipple factor is a multiplier that is used when drawing stipple patterns.

Example (*Figure: Example of using the SetStippleFactor method*)

```
OEPen pen = new OEPen();
pen.SetLineStipple(OEStipple.ShortDash);
pen.SetStippleFactor(2);
```

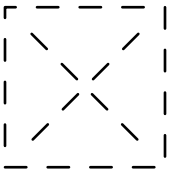


Fig. 209: **Example of using the SetStippleFactor method**

See also

- *OEPen.GetStippleFactor* method
- *OEPen.GetLineStipple* method

4.1.52 OEPrepareDepictionOptions

```
class OEPrepareDepictionOptions
```

This class represents *OEPrepareDepictionOptions* that stores the following parameters that are used when preparing a molecule for depiction.

Property	Get method	Set method	Corresponding namespace / type
re-generating 2D coordinates	*GetClearCoords*	*SetClearCoords*	boolean
adding depiction hydrogens	*GetAddDepictionHy*	*SetAddDepictionHy*	boolean
perceiving MDL bond stereo (wedge/hash)	*GetPerceiveBondSt*	*SetPerceiveBondSt*	boolean
suppressing explicit (non-depiction) hydrogens	*GetSuppressHydrog*	*SetSuppressHydrog*	boolean
depiction orientation	*GetDepictOrientat*	*SetDepictOrientat*	*OEDepictOrientation*
algorithmic macrocycle ring depiction	*GetOptimizeMacroc*	*SetOptimizeMacroc*	boolean

See also

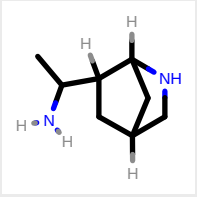
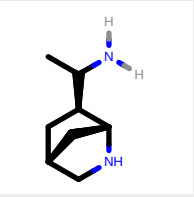
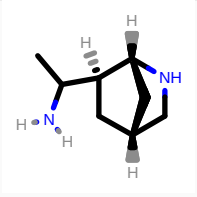
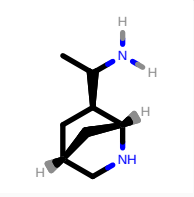
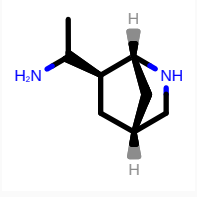
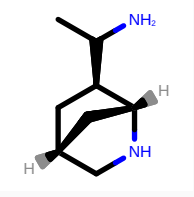
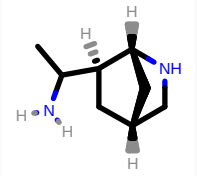
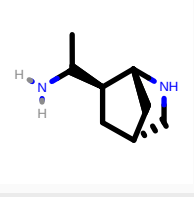
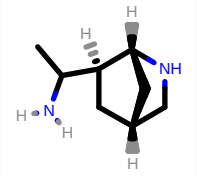
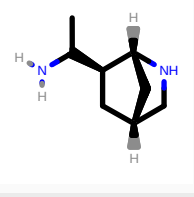
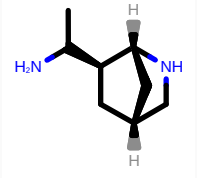
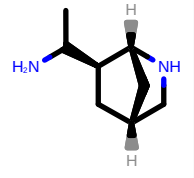
- *OEPrepareDepiction* function

See the effect of the above parameters in *Table: Depictions generated with various parameter combinations* when depiction are generated from the following input:

SMILES
: Parsing the "CC([C@@H]1C[C@@H]2C[C@H]1NC2)N(H)(H)" string with three explicit hydrogens necessary to define tetrahedral stereo centers and two additional explicit hydrogens.

MDL
: Reading an MDL file with 2D coordinates and pre-assigned wedge/hash bonds. It also includes two explicit hydrogens.

Table 11: **Depictions generated with various parameter combinations**

index	clear coordinates	suppress explicit H	add depict H	perceive wedge/hash	from SMILES	from MDL
(1)	F	F	F	F		
(2)	F	F	F	**T**		
(3)	F	F	**T**	F		
(4)	F	F	**T**	**T**		
(5)	F	**T**	F	F		
(6)	F	**T**	F	**T**		
(7)	F	**T**	**T**	F		
(8)	F	**T**	**T**	**T**		
(9)	**T**	F	F	F		
(10)	**T**	F	F	**T**		
(11)	**T**	F	**T**	F		
(12)	**T**	F	**T**	**T**		
(13)	**T**	**T**	F	F		
(14)	**T**	**T**	F	**T**		
(15)	**T**	**T**	**T**	F		
(16)	**T**	**T**	**T**	**T**		

Comments:

(1)
MDL = no op; SMILES = generating 2D coordinates is necessary

(4)
MDL = keeping original 2D coordinates, adding depiction hydrogens, re-perceiving wedge/hash bonds; SMILES = generating 2D coordinates, depiction hydrogens are kept; perceiving wedge/hash bonds

(8)
Default parameters; MDL = keeping original 2D coordinates, re-perceiving all other information; SMILES = generating 2D coordinates, regenerating all other information;

(10)
MDL = regenerating 2D coordinates and re-perceiving wedge/hash bonds (since no depiction hydrogens are added the wedge/hash bonds are on ring bonds); SMILES generating 2D coordinates and perceiving wedge/hash bonds;

(12)
Regenerating all information and keeping all explicit hydrogens

(16)
Regenerating all information; the depiction of identical molecules imported from SMILES and MDL will be identical

Constructors

```
OEPrepareDepictionOptions(bool clearcoords = false,
                          bool suppressH = true)
```

Default constructor that initializes an *OEPrepareDepictionOptions* object with the following properties:

clearcoords
See the *OEPrepareDepictionOptions.SetClearCoords* method.

suppressH
See the *OEPrepareDepictionOptions.SetSuppressHydrogens* method.

Table 12: **Default parameters of the OEPrepareDepictionOptions class**

Property	Default value
re-generating 2D coordinates	false
adding depiction hydrogens	true
perceiving MDL bond stereo (wedge/hash)	true
suppressing explicit (non-depiction) hydrogens	true
depiction orientation	*OEDepictOrientation.Default*
algorithmic macrocycle ring depiction	false

```
OEPrepareDepictionOptions(const OEPrepareDepictionOptions& rhs)
```

Copy constructor.

GetAddDepictionHydrogens

```
bool GetAddDepictionHydrogens() const
```

Returns whether the depiction hydrogens are kept/added.

See also

- *OEPrepareDepictionOptions.SetAddDepictionHydrogens* method

GetClearCoords

```
bool GetClearCoords() const
```

Returns whether the 2D coordinates of a molecule are re-calculated.

See also

- *OEPrepareDepictionOptions.SetClearCoords* method

GetDepictOrientation

```
unsigned int GetDepictOrientation() const
```

Returns the preferred orientation of the 2D coordinates. The return value is taken from the *OEDepictOrientation* namespace.

See also

- *OEPrepareDepictionOptions.SetDepictOrientation* method

GetOptimizeMacrocycles

```
bool GetOptimizeMacrocycles() const
```

Returns whether the algorithmic macrocycle ring depiction is enabled.

See also

- *OEPrepareDepictionOptions.SetOptimizeMacrocycles* method

GetPerceiveBondStereo

```
bool GetPerceiveBondStereo() const
```

Returns whether the hash/wedge bonds are re-assigned.

See also

- *OEPrepareDepictionOptions.SetPerceiveBondStereo* method

GetSuppressHydrogens

```
bool GetSuppressHydrogens() const
```

Returns whether the explicit (non-depiction) hydrogens are suppressed.

See also

- *OEPrepareDepictionOptions.SetSuppressHydrogens* method

SetAddDepictionHydrogens

```
void SetAddDepictionHydrogens(bool)
```

Sets whether depiction hydrogens are kept/added.

See also

- *OEAddDepictionHydrogens* function
- *OEPrepareDepictionOptions.GetAddDepictionHydrogens* method

SetClearCoords

```
void SetClearCoords(bool clearcoords)
```

Sets whether the 2D coordinates of the molecule are re-calculated. If the molecule has no 2D coordinates, then the 2D coordinates are always calculated by calling the *OEDepictCoordinates* regardless of this setting.

See also

- *OEDepictCoordinates* function
- *OEPrepareDepictionOptions.GetClearCoords* method

SetDepictOrientation

```
void SetDepictOrientation(unsigned int orientation)
```

Sets the preferred orientation of the 2D coordinates.

orientation
This value has to be from the *OEDepictOrientation* namespace.

See also

- *OEPrepareDepictionOptions.GetDepictOrientation* method
- *OEDepictOrientation* namespace

SetOptimizeMacrocycles

```
bool SetOptimizeMacrocycles(bool)
```

Sets whether to use the algorithmic layout for macrocycle ring systems. When enabled, *OEPrepareDepiction* will attempt to generate improved algorithmic 2D layout for macrocycles rather than using the template lookup or polygon fallback. If the algorithm cannot handle a particular ring topology, coordinate generation fails and *OEPrepareDepiction* returns `false`. It is the caller's responsibility to retry without this option enabled if a fallback depiction is required.

See also

- *OEPrepareDepictionOptions.GetOptimizeMacrocycles* method
- *OEPrepareDepiction* function

SetPerceiveBondStereo

```
bool SetPerceiveBondStereo(bool)
```

Sets whether to re-assign the wedge and hash bonds by calling the `OEMDLPerceiveBondStereo` function.

See also

- `OEMDLPerceiveBondStereo` function in the **OEChem TK** manual
- *OEPrepareDepictionOptions.GetPerceiveBondStereo* method

SetSuppressHydrogens

```
void SetSuppressHydrogens(bool)
```

Sets whether the explicit (non-depiction) hydrogens are suppressed in the molecule. Only hydrogens that are necessary to faithfully represent tetrahedral stereo-chemistry will be kept.

See also

- `OESuppressHydrogens` function in the **OEChem TK** manual
- *`OEPrepareDepictionOptions.GetSuppressHydrogens`* method

4.1.53 OEReport

```
class OEReport
```

This class represents a layout manager that allows generation of multi-page images with two types of page layouts:

body
: The *body* page layout provides a convenient way to position information in the middle of the page relative to the page footers and headers. For example, see the 1^{st} page in *Figure: Schematic representation of documents that can be created using the OEReport class*.

grid
: The *grid* page layout provides a convenient way to organize information (such as text and molecule depiction) into rows and columns. For example, see the 2^{nd} and 3^{rd} pages in *Figure: Schematic representation of documents that can be created using the OEReport class*.

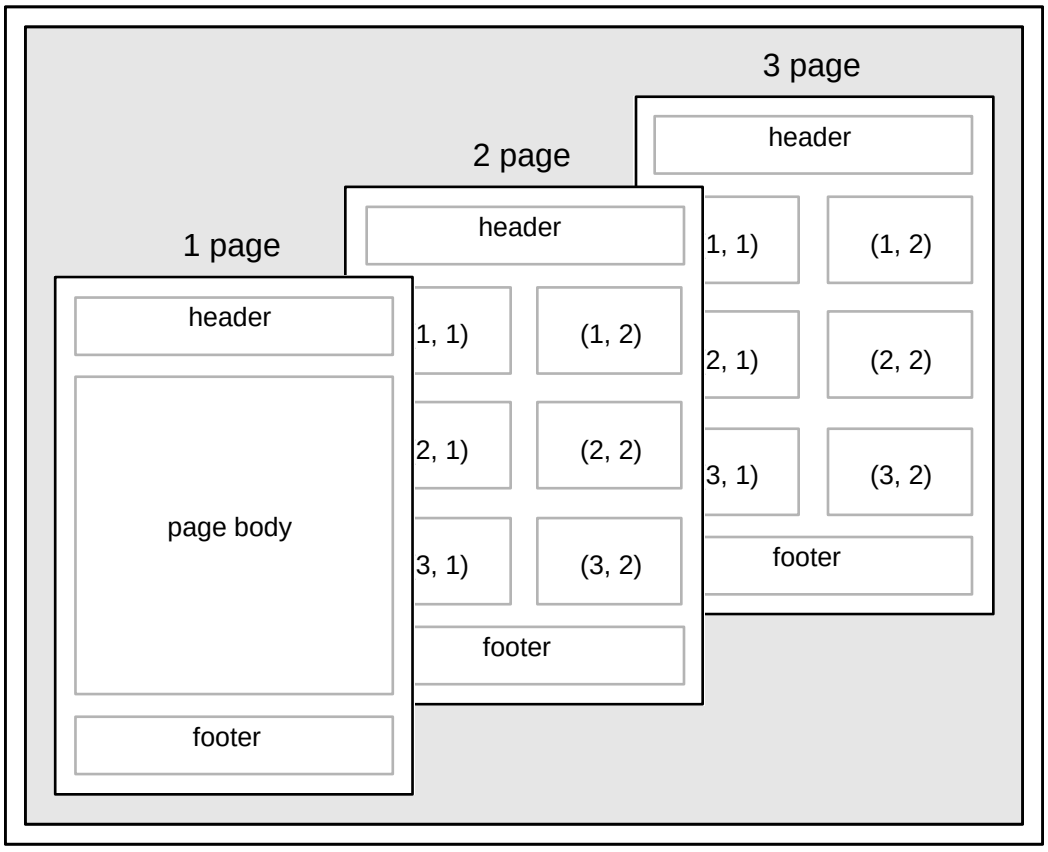


Fig. 210: **Schematic representation of documents that can be created using the OEReport class**

See also

- *Multi Page Reports* section
- *Depicting Molecules in Multi-Page* example
- *Depicting MDL Query Substructure Search Hits* example
- *OEReportOptions* class
- *OEWriteReport* function
- *OEWriteReportToString* function

Constructors

```
OEReport(const OEReportOptions& opts)
```

Initializes an *OEReport* object using the given options.

opts
The *OEReportOptions* object that stores parameters determining the layout of the *body* and the *grid* pages of the report.

```
OEReport(unsigned int rowsperpage, unsigned int colsperpage)
```

Initializes an *OEReport* object with the given parameters.

rowsperpage
The number of rows on each *grid* layout page.

colsperpage
The number of columns on each *grid* layout page.

Note

When a report is initialized with only the number of rows and columns per page parameter, all other parameters defining the layout of the pages will be initialized from the default constructor of the *OEReportOptions* class.

See also

- *OEReportOptions* class

GetBodies

```
OESystem::OEIterBase<OEImageBase>* GetBodies()
```

Returns an iterator over all page bodies that have been created by calling the *OEReport.NewBody* method.

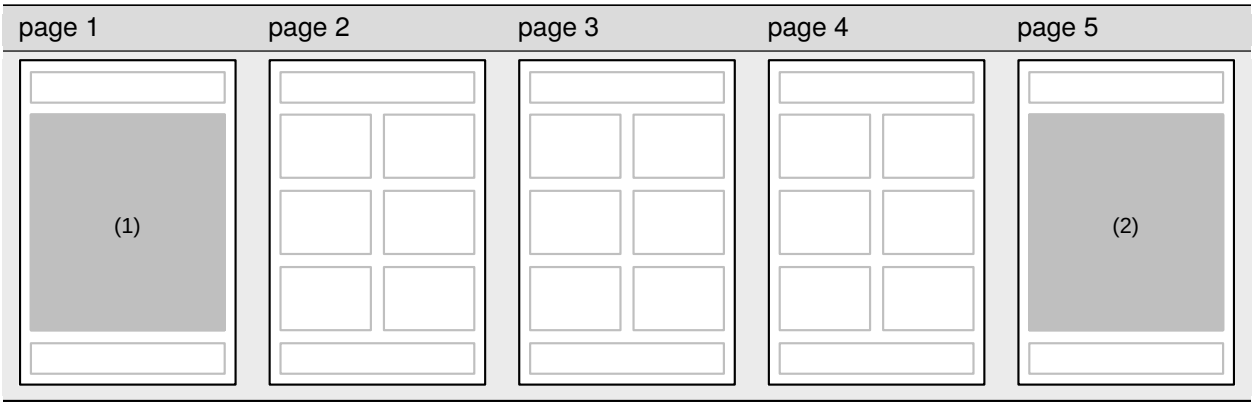
Note

If no page has been created by the *OEReport.NewBody* method, then the *OEReport.GetBodies* method returns an empty iterator.

See example in *Figure: Example of accessing all page bodies of the report*, where *body* areas in the 1^{st} and 5^{th} page are returned.

```
for (OEImageBase body : report.GetBodies())
```

Table 13: **Example of accessing all page bodies of the report**. (The numbers inside the cells represent the order in which they are returned)



See also

- *OEReport.GetBody* method
- *OEReport.NewBody* method

GetBody

```
OEImageBase* GetBody(unsigned int p)
```

Returns the *OEImageBase* pointer of the body area of the p^{th} page.

p
: This value has to be in the range from *1* to *OEReport.NumPages()*.

Note

If the p^{th} page of the report was not created by the *OEReport.NewBody* method or the p page index is out of range, then the *OEReport.GetBody* method returns a NULL pointer.

See also

- *OEReport.GetBodies* method

- *OEReport.NewBody* method

GetCell

```
OEImageBase* GetCell(unsigned int p, unsigned int r, unsigned int c)
```

Returns the *OEImageBase* pointer for the cell positioned in the r^{th} row and c^{th} columns on the p^{th} page.

p
: This value has to be in the range from *1* to *OEReport.NumPages()*.

r
: This value has to be in the range from *1* to *OEReport.NumRowsPerPage()*.

c
: This value has to be in the range from *1* to *OEReport.NumColsPerPage()*.

Note

If the cell positioned in the r^{th} row and c^{th} columns on the p^{th} page has not been created by the *OEReport.NewCell* method, then the *OEReport.GetCell* method returns a NULL pointer.

See also

- *OEReport.GetCells* method
- *OEReport.NewCell* method

GetCellHeight

```
double GetCellHeight() const
```

Returns the height of the cells on the grid page of the *OEReport* object.

GetCellWidth

```
double GetCellWidth() const
```

Returns the width of the cells on the grid page of the *OEReport* object.

GetCells

```
OESystem::OEIterBase<OEImageBase>* GetCells(unsigned int page = 0u,
                                            unsigned int row = 0u,
                                            unsigned int col = 0u)
```

Returns an iterator over cells that have been created by calling the *OEReport.NewCell* method. The cells are returned in the order in which they have been created *i.e.* from left to right and top to bottom order on each grid page.

page
: This value has to be in the range from *0* to *OEReport.NumPages()*.

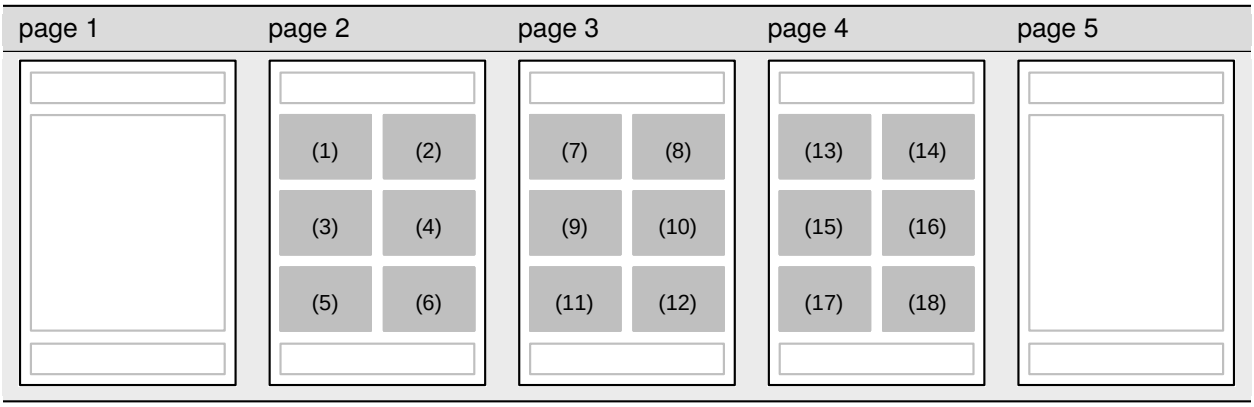
row
: This value has to be in the range from *0* to *OEReport.NumRowsPerPage()*.

col
: This value has to be in the range from *0* to *OEReport.NumColsPerPage()*.

Example (*Figure: Example of accessing all cells of the report*)

```
for (OEImageBase cell : report.GetCells())
```

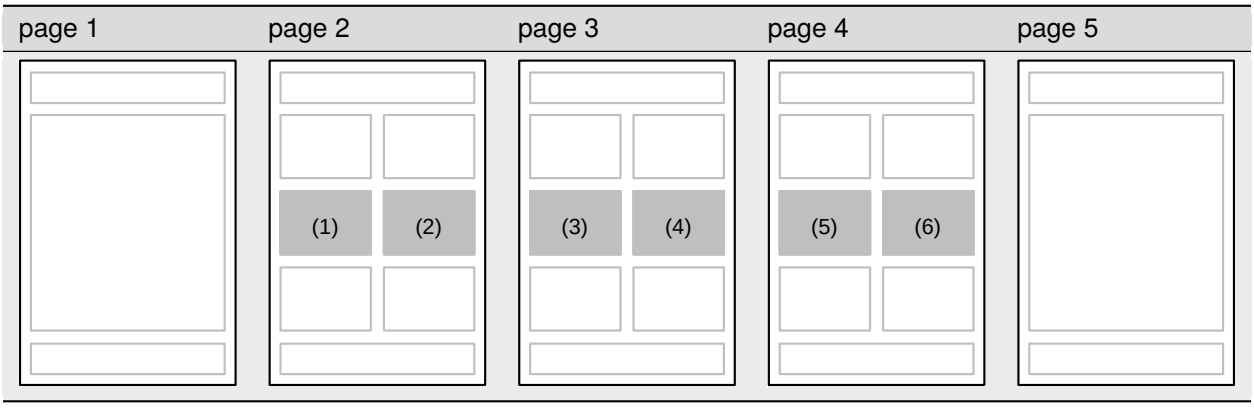
Table 14: **Example of accessing all cells of the report**. (The numbers inside the cells represent the order in which they are returned)



Example (*Figure: Example of accessing all cells on the given row on each page*)

```
int row = 2;
for (OEImageBase cell : report.GetCells(0, row, 0))
```

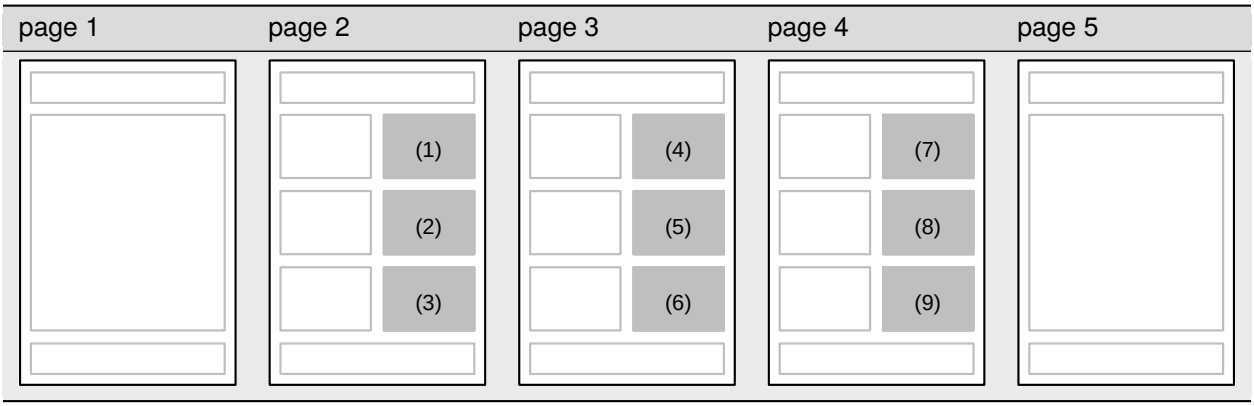
Table 15: **Example of accessing all cells on the given row on each page**. (The numbers inside the cells represent the order in which they are returned)



Example (*Figure: Example of accessing all cells on the given column on each page*)

```
int col = 2;
for (OEImageBase cell : report.GetCells(0, 0, col))
```

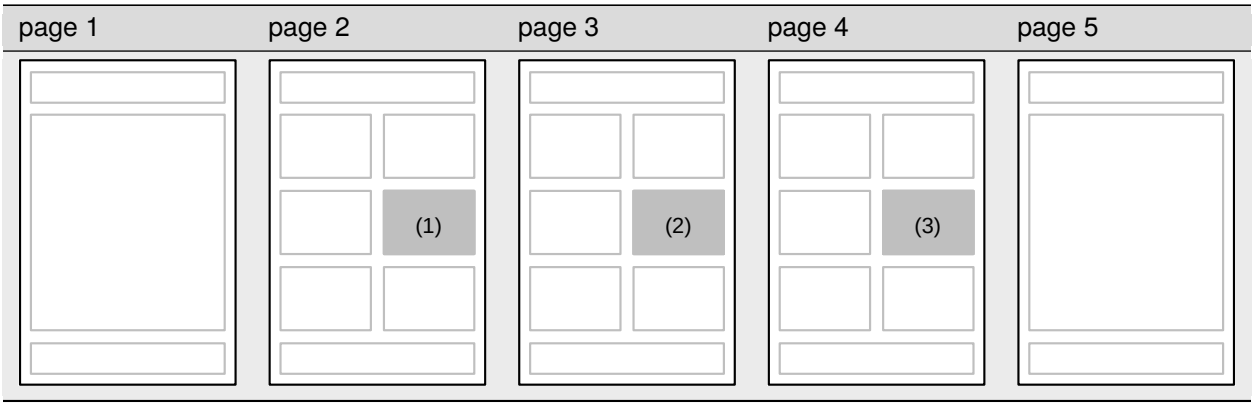
Table 16: **Example of accessing all cells on the given column on each page**. (The numbers inside the cells represent the order in which they are returned)



Example (*Figure: Example of accessing all cells on the given row and column on each page*)

```
int row = 2;
int col = 2;
for (OEImageBase cell : report.GetCells(0, row, col))
```

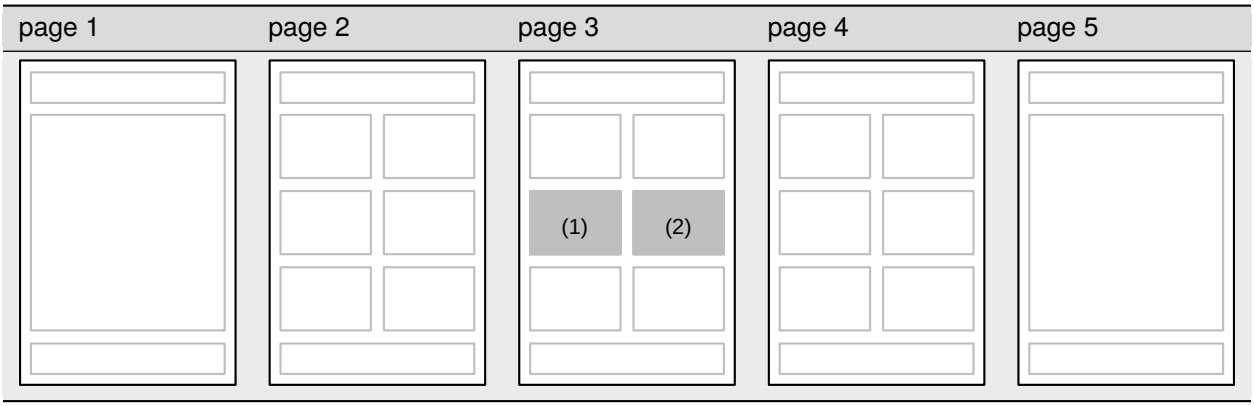
Table 17: **Example of accessing all cells on the given row and column on each page**. (The numbers inside the cells represent the order in which they are returned)



Example (*Figure: Example of accessing all cells on the given row and page*)

```
int page = 3;
int row = 2;
for (OEImageBase cell : report.GetCells(page, row, 0))
```

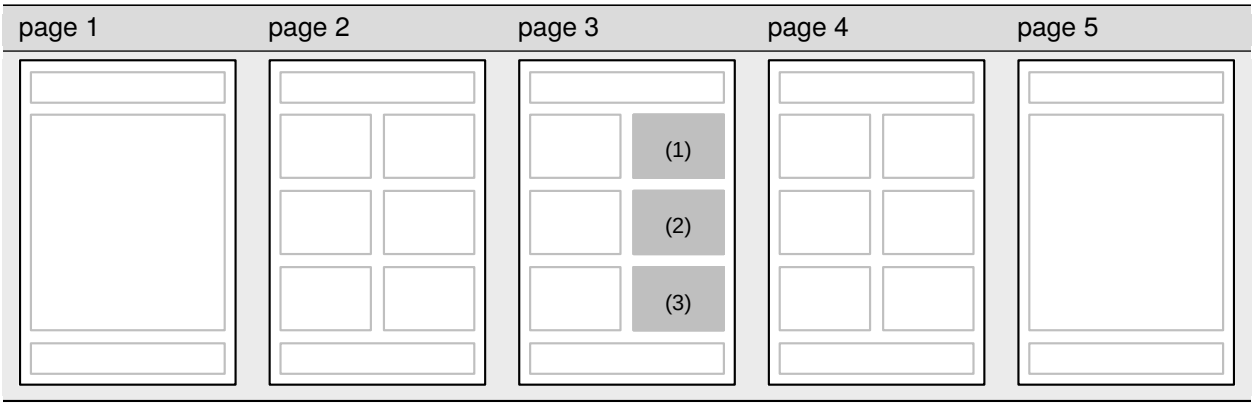
Table 18: **Example of accessing all cells on the given row and page**. (The numbers inside the cells represent the order in which they are returned)



Example (*Figure: Example of accessing all cells on the given column and page*)

```
int page = 3;
int col = 2;
for (OEImageBase cell : report.GetCells(page, 0, col))
```

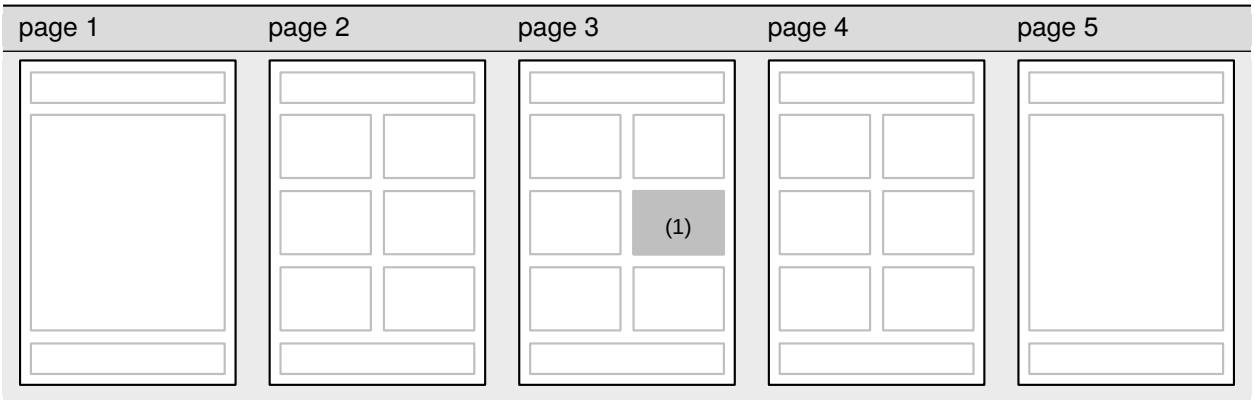
Table 19: **Example of accessing all cells on the given column and page**. (The numbers inside the cells represent the order in which they are returned)



Example (*Figure: Example of accessing a cell on the given row, column and page*)

```
int page = 3;
int row = 2;
int col = 2;
for (OEImageBase cell : report.GetCells(page, row, col))
```

Table 20: **Example of accessing a cell on the given row, column and page**. (The numbers inside the cells represent the order in which they are returned)



See also

- *OEReport.GetCell* method
- *OEReport.NewCell* method

GetFooter

```
OEImageBase* GetFooter(unsigned int p)
```

Returns the *OEImageBase* pointer of the footer on the p^{th} page.

p
: This value has to be in the range from *1* to *OEReport.NumPages()*.

Note

If the *OEReport* object was initialized with footer height *0.0* or the *p* page index is out of range, then the *OEReport.GetFooter* method returns a NULL pointer.

See also

- *OEReport.GetFooterHeight* method
- *OEReport.GetFooterWidth* method
- *OEReport.GetFooters* methods

GetFooterHeight

```
double GetFooterHeight() const
```

Returns the height of the footer at the bottom of each page.

See also

- *OEReport.GetFooterWidth* method

GetFooterWidth

```
double GetFooterWidth() const
```

Returns the width of the footer at the bottom of each page.

See also

- *OEReport.GetFooterHeight* methods

GetFooters

```
OESystem::OEIterBase<OEImageBase>* GetFooters()
```

Returns an iterator over all footers of the *OEReport* object.

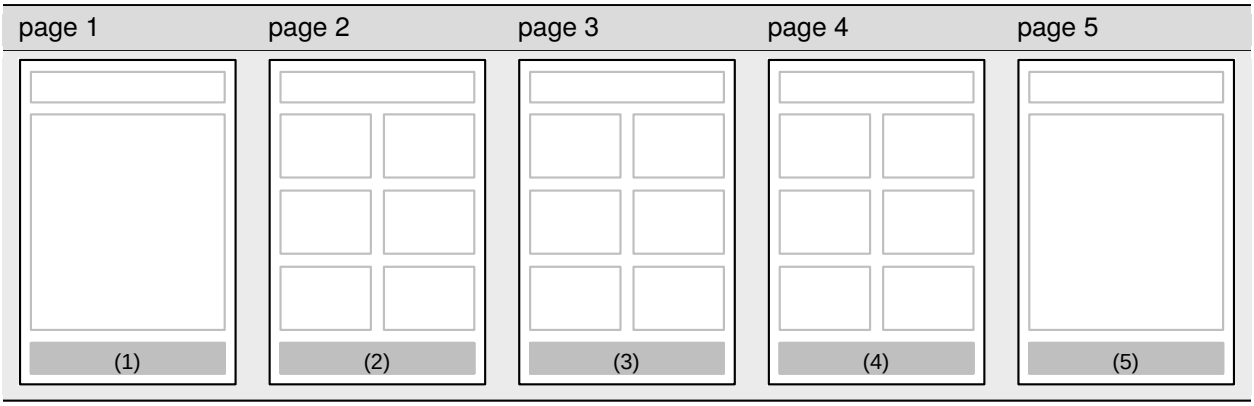
Note

If the *OEReport* object was initialized with footer height *0.0*, then the *OEReport.GetFooters* method returns an empty iterator.

Example (*Figure: Example of accessing all page footers of the report*)

```
for (OEImageBase footer : report.GetFooters())
```

Table 21: **Example of accessing all page footers of the report**. (The numbers inside the footer areas represent the order in which they are returned)



See also

- *OEReport.GetFooter* method
- *OEReport.GetFooterHeight* method
- *OEReport.GetFooterWidth* method

GetHeader

```
OEImageBase* GetHeader(unsigned int p)
```

Returns the *OEImageBase* pointer of the header on the p^{th} page.

p
: This value has to be in the range from *1* to *OEReport.NumPages()*.

Note

If the *OEReport* object was initialized with header height *0.0* or the *p* page index is out of range, then the *OEReport.GetHeader* method returns a NULL pointer.

See also

- *OEReport.GetHeaderHeight* method
- *OEReport.GetHeaderWidth* method
- *OEReport.GetHeaders* method

GetHeaderHeight

```
double GetHeaderHeight() const
```

Returns the height of the header at the top of each page.

See also

- *OEReport.GetHeaderWidth* method

GetHeaderWidth

```
double GetHeaderWidth() const
```

Returns the width of the header at the top of each page.

See also

- *OEReportOptions.GetHeaderHeight* methods

GetHeaders

```
OESystem::OEIterBase<OEImageBase>* GetHeaders()
```

Returns an iterator over all headers of the *OEReport* object.

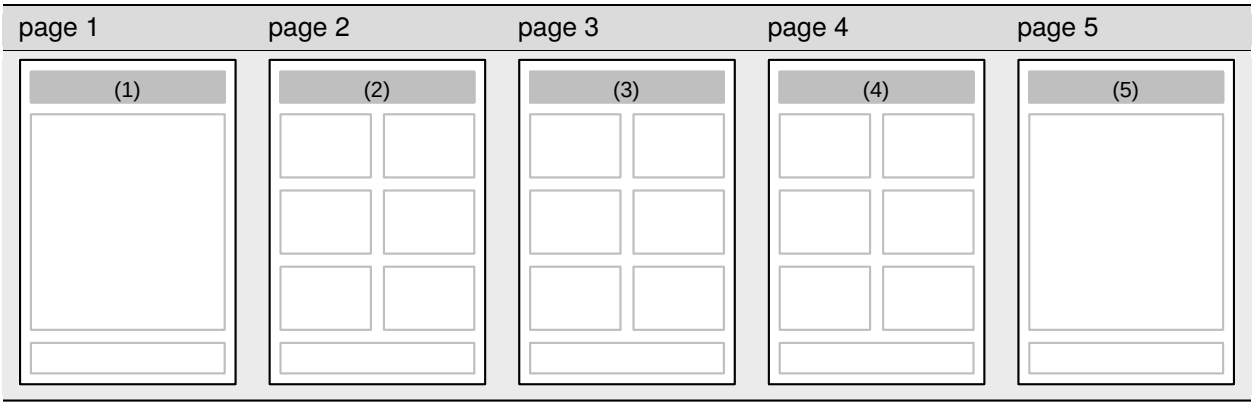
Note

If the *OEReport* object was initialized with header height *0.0*, then the *OEReport.GetHeaders* method returns an empty iterator.

Example (*Figure: Example of accessing all page headers of the report*)

```
for (OEImageBase header : report.GetHeaders())
```

Table 22: **Example of accessing all page headers of the report**. (The numbers inside the header areas represent the order in which they are returned)



See also

- *OEReport.GetHeader* method
- *OEReport.GetHeaderHeight* method
- *OEReport.GetHeaderWidth* method

GetOptions

```
const OEReportOptions& GetOptions() const
```

Returns the options used to initialize the *OEReport* object.

See also

- *OEReportOptions* class

GetPage

```
OEImageBase* GetPage(unsigned int p)
```

Returns the *OEImageBase* pointer of the p^{th} page.

p
: This value has to be in the range from *1* to *OEReport.NumPages()*.

Note

If p page index is out of range, then the *OEReport.GetPage* method returns a NULL pointer.

See also

- *OEReport.GetPages* method

GetPageHeight

```
double GetPageHeight() const
```

Returns the height of the pages of the report.

See also

- *OEReportOptions.SetPageHeight* method

GetPageWidth

```
double GetPageWidth() const
```

Returns the width of the pages of the report.

See also

- *OEReportOptions.SetPageWidth* method

GetPages

```
OESystem::OEIterBase<OEImageBase>* GetPages()
```

Returns an iterator over all the pages of the *OEReport* object.

See also

- *OEReport.GetPage* method

IsBodyPage

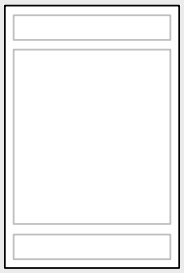
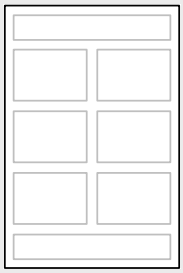
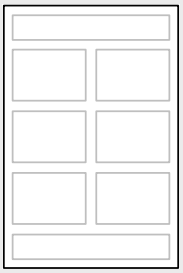
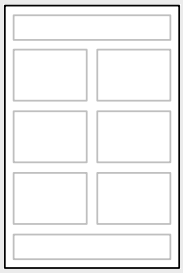
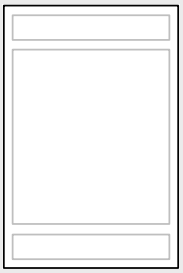
```
bool IsBodyPage(unsigned int p) const
```

Returns whether the p^{th} page of the report is created by calling the *OEReport.NewBody* method.

p
: This value has to be in the range from *1* to *OEReport.NumPages()*.

For example, the *OEReport.IsBodyPage* method returns *true* for the 1^{st} and 5^{th} pages of the report shown in *Figure: Example of report layout*.

Table 23: **Example of report layout**

	page 1	page 2	page 3	page 4	page 5
					
IsBodyPage	true	false	false	false	true

See also

- *OEReport.IsGridPage* method
- *OEReport.NewBody* method

IsGridPage

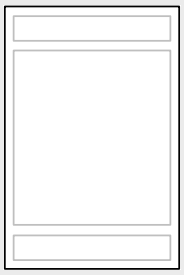
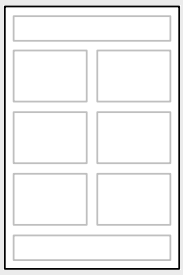
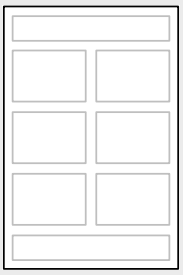
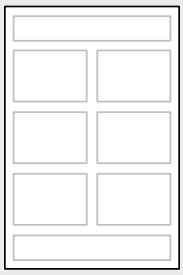
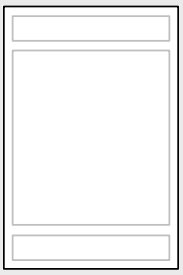
```
bool IsGridPage(unsigned int p) const
```

Returns whether the p^{th} page of the report is created by calling the *OEReport.NewCell* method.

p
: This value has to be in the range from *1* to *OEReport.NumPages()*.

For example, the *OEReport.IsGridPage* method returns *true* for the 2^{nd}, 3^{rd}, and 4^{th} pages of the report shown in *Figure: Example of report layout*.

Table 24: **Example of report layout**

	page 1	page 2	page 3	page 4	page 5
					
IsGridPage	false	true	true	true	false

See also

- *`OEReport.IsBodyPage`* method
- *`OEReport.NewCell`* method

NewBody

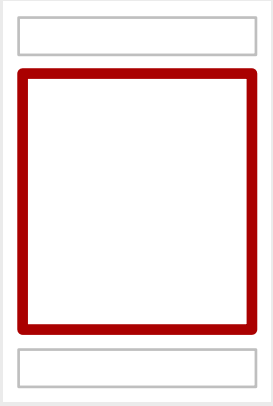
```
OEImageBase* NewBody()
```

Creates a new *body* layout page and returns the area between the header and the footer. See example in *Figure: Example of creating a new body page*.

Note

The height and width of the returned area depend on the options with which the *OEReport* object was constructed: page margins, gap between the cells, and the height of the header and footer.

Table 25: **Example of creating a new body page** (The area returned by the NewBody method is highlighted by red color)

page 1	page 2	page 3
		

See also

- *OEReport.IsBodyPage* method
- *OEReport.GetBody* method
- *OEReport.GetBodies* method

NewCell

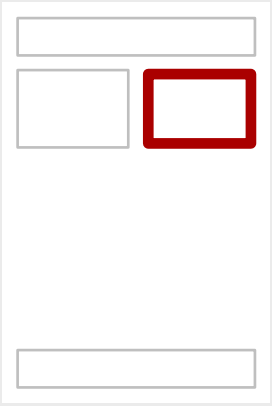
```
OEImageBase* NewCell()
```

Creates and returns a new cell. If the last page of the report is a *grid* page, then a new cell is created from left to right and top to bottom order. If no more cells are left on the *grid* page or the last page is a *body* page, then a new *grid* page is created before returning the first cell of this new page. See example in *Figure: Example of creating a new cell*.

Note

The cells of a grid page are equal-sized and evenly spaced. The height and width of the cells depend on the options with which the *OEReport* object was constructed: the number of rows and columns per page, page margins, gap between the cells, and the height of the header and footer.

Table 26: **Example of creating a new cell** (The area i.e. the cell returned by the NewCell method is highlighted by red color)

page 1	page 2	page 3
		

See also

- *OEReport.IsGridPage* method
- *OEReport.GetCell* method
- *OEReport.GetCells* method

NumBodyPages

```
unsigned int NumBodyPages() const
```

Returns the number of *body* pages of the *OEReport* object, *i.e.* the number of pages for which the *OEReport.IsBodyPage* method returns *true*.

See also

- *OEReport.IsBodyPage* method
- *OEReport.NumGridPages* method
- *OEReport.NumPages* method

NumColsPerPage

```
unsigned int NumColsPerPage() const
```

Returns the number of columns on each *grid* page of the *OEReport* object.

See also

- *OEReport.NumRowsPerPage* method

NumGridPages

```
unsigned int NumGridPages() const
```

Returns the number of *grid* pages of the *OEReport* object, *i.e.* the number of pages for which the *OEReport.IsGridPage* method returns *true*.

See also

- *OEReport.IsGridPage* method
- *OEReport.NumBodyPages* method
- *OEReport.NumPages* method

NumPages

```
unsigned int NumPages() const
```

Returns the number of pages of the *OEReport* object. This is the sum of the numbers returned by the *OEReport.NumBodyPages* and *OEReport.NumGridPages* methods.

See also

- *OEReport.NumBodyPages* method
- *OEReport.NumGridPages* method

NumRowsPerPage

```
unsigned int NumRowsPerPage() const
```

Returns the number of rows on each *grid* page of the *OEReport* object.

See also

- *OEReport.NumColsPerPage* method

4.1.54 OEReportOptions

```
class OEReportOptions
```

This class represents the *OEReportOptions* class that encapsulates properties that determine the layout of a *OEReport* object.

See *Figure: Schematic representation of documents that can be created using the OEReport class*.

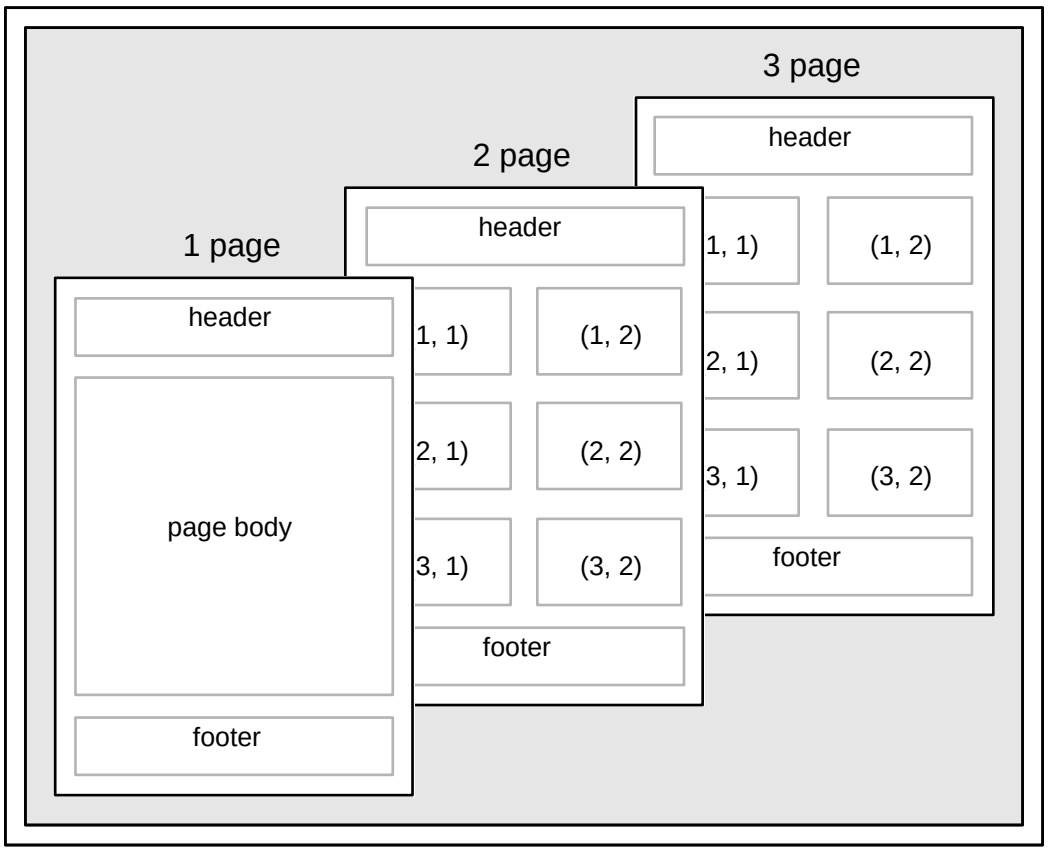


Fig. 211: **Schematic representation of documents that can be created using the OEReport class**

The *OEReportOptions* class stores the following properties:

Property	Get method	Set method	Corresponding namespace / class / type
rows per page	*NumRowsPerP*	set upon construction	integer greater than 0
columns per page	*NumColsPerP*	set upon construction	integer greater than 0
page width	*GetPageWidt*	*SetPageWidth* / *SetPageSize* / *SetPageOrientation*	floating point number / *OEPageSize* / *OEPageOrientation*
page height	*GetPageHeig*	*SetPageHeight* / *SetPageSize* / *SetPageOrientation*	floating point number / *OEPageSize* / *OEPageOrientation*
page margins	*GetPageMarg*	*SetPageMargin* / *SetPageMargins*	*OEMargin*
cell gap	*GetCellGap*	*SetCellGap*	floating point number
header height	*GetHeaderHe*	*SetHeaderHeight*	floating point number
header width	*GetHeaderWi*	N/A	floating point number
footer height	*GetFooterHe*	*SetFooterHeight*	floating point number
footer width	*GetFooterWi*	N/A	floating point number

See also

- *OEReport* class

Constructors

```
OEReportOptions(unsigned int rowsperpage = 3u,
                unsigned int colsperpage = 2u)
```

rowsperpage
The number of rows on each *grid* page of the report.

colsperpage
The number of columns on each *grid* page of the report.

Note

Both the number of rows and columns per page have to be greater than 0.

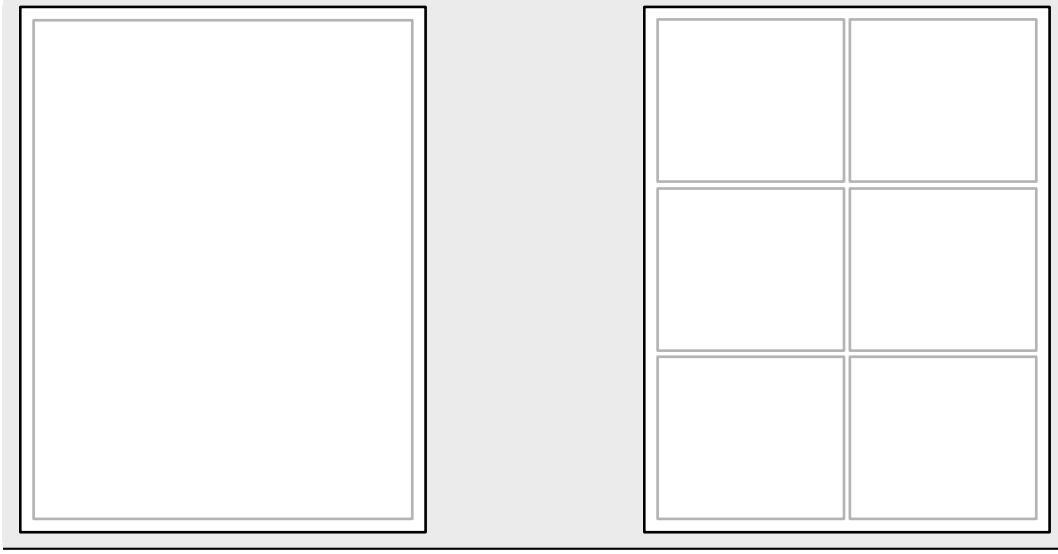
Table 27: **Default parameters of OEReportOptions**

Property	Default value
rows per page	3
columns per page	2
page width	*OEGetPageWidth* (*OEPageSize.US_Letter*)
page height	*OEGetPageHeight* (*OEPageSize.US_Letter*)
margins	20.0 (for each margins: *top*, *bottom*, *left* and *right*)
cell gap	5.0
header height	0.0 (no header is generated by default)
footer height	0.0 (no footer is generated by default)

Example (*Figure: Example of report generation with default options*)

```
OEReportOptions opts = new OEReportOptions(3, 2);
```

Table 28: **Example of report generation with default options** (The scale of the shown pages is 1:4)



```
OEReportOptions(const OEReportOptions& rhs)
```

Copy constructor.

GetCellGap

```
double GetCellGap() const
```

Returns the space between the cells of the report.

See also

- *OEReportOptions.SetCellGap* method

GetCellHeight

```
double GetCellHeight() const
```

Returns the height of the cells on the *grid* pages of the report.

Note

The height of the cells depends on the number of rows per page, the top and bottom margins, header and footer heights, and the gap between the cells

GetCellWidth

```
double GetCellWidth() const
```

Returns the width of the cells on the *grid* pages of the report.

Note

The width of the cells depends on the number of columns per page, the left and right margins, and the gap between the cells.

GetFooterHeight

```
double GetFooterHeight() const
```

Returns the height of the footer at the bottom of each page.

See also

- *OEReportOptions.SetFooterHeight* method

GetFooterWidth

```
double GetFooterWidth() const
```

Returns the width of the footer at the bottom of each page.

GetHeaderHeight

```
double GetHeaderHeight() const
```

Returns the height of the header at the top of each page.

See also

- *OEReportOptions.SetHeaderHeight* method

GetHeaderWidth

```
double GetHeaderWidth() const
```

Returns the width of the header at the top of each page.

GetPageHeight

```
double GetPageHeight() const
```

Returns the height of the page of the report.

See also

- *OEReportOptions.SetPageHeight* method

GetPageMargin

```
double GetPageMargin(unsigned int margin) const
```

Returns the size of a specific page margin of the report.

margin
: This value has to be from the *OEMargin* namespace.

See also

- *OEReportOptions.SetPageMargin* method
- *OEReportOptions.SetPageMargins* method

GetPageWidth

```
double GetPageWidth() const
```

Returns the width of the page of the report.

See also

- *OEReportOptions.SetPageWidth* method

NumColsPerPage

```
unsigned int NumColsPerPage() const
```

Returns the number of columns on each *grid* page of the report.

See also

- *OEReportOptions.NumRowsPerPage* method

NumRowsPerPage

```
unsigned int NumRowsPerPage() const
```

Returns the number of rows on each *grid* page of the report.

See also

- *OEReportOptions.NumColsPerPage* method

SetCellGap

```
void SetCellGap(double size)
```

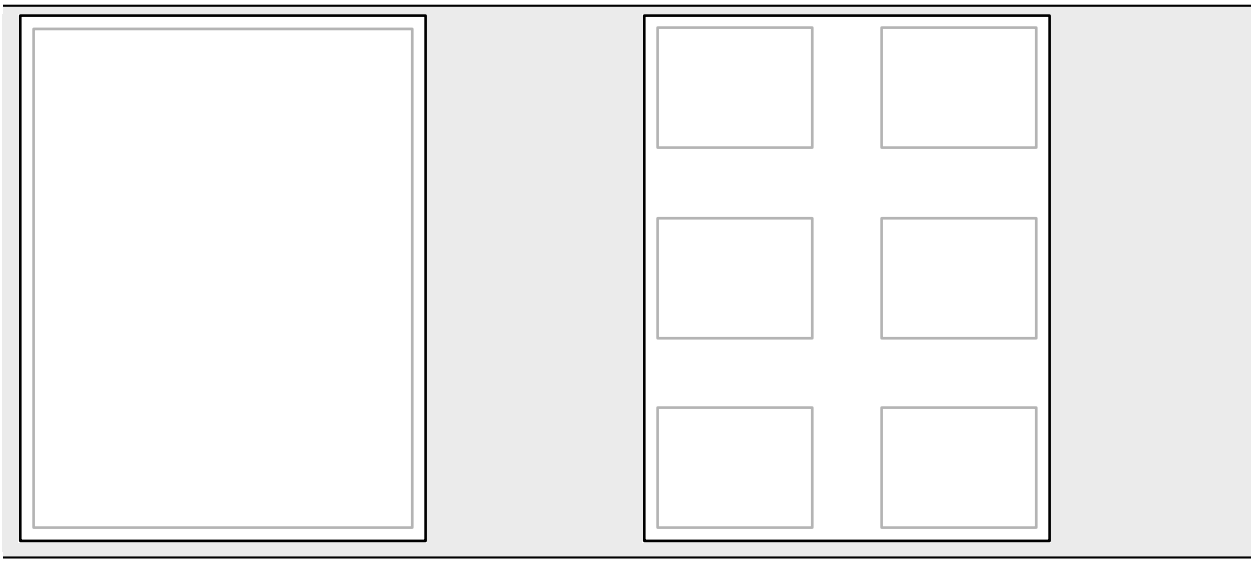
Sets the space between the cells of the report.

size
This value has to be a non-negative number.

Example (*Figure: Example of using the SetCellGap method*)

```
OEReportOptions opts = new OEReportOptions(3, 2);
opts.SetCellGap(100);
```

Table 29: **Example of using the SetCellGap method** (The scale of the shown pages is 1:4)



See also

- *OEReportOptions.GetCellGap* method

SetFooterHeight

```
void SetFooterHeight(double height)
```

Sets the height of the footer at the bottom of each page.

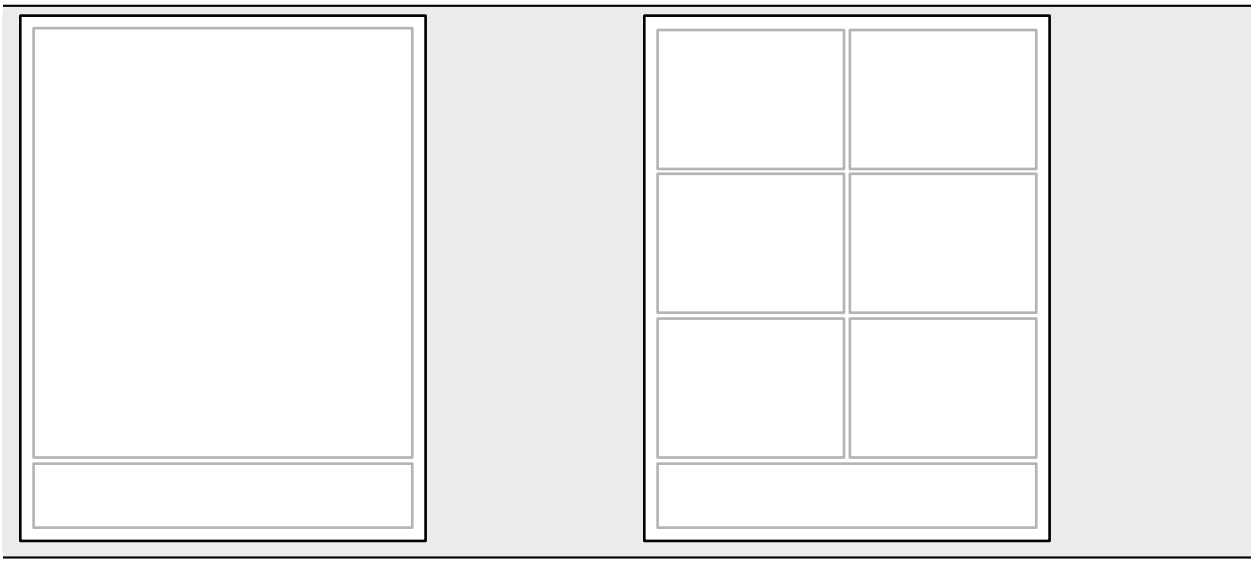
height
: This value has to be a non-negative number.

Note

0.0 means that a footer is **not** going to be created on the pages.

Example (*Figure: Example of using the SetFooterHeight method*)

Table 30: **Example of using the SetFooterHeight method** (The scale of the shown pages is 1:4)



See also

- *OEReportOptions.GetFooterHeight* method

SetHeaderHeight

```
void SetHeaderHeight(double height)
```

Sets the height of the header at the top of each page.

height
: This value has to be a non-negative number.

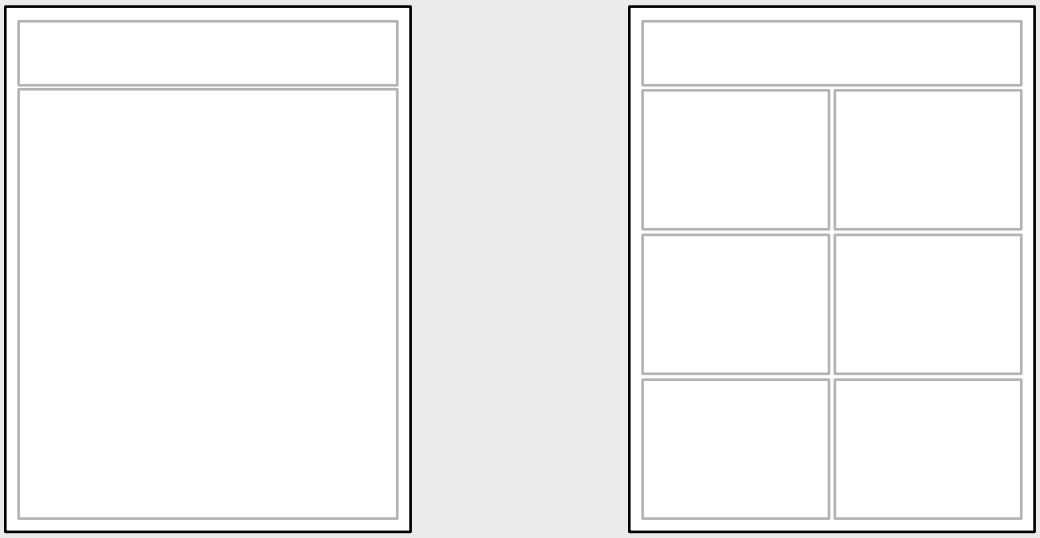
Note

0.0 means that a header is **not** going to be created on the pages.

Example (*Figure: Example of using the SetHeaderHeight method*)

```
OEReportOptions opts = new OEReportOptions(3, 2);
opts.SetHeaderHeight(100);
```

Table 31: **Example of using the SetHeaderHeight method** (The scale of the shown pages is 1:4)



See also

- *OEReportOptions.GetHeaderHeight* method

SetPageHeight

```
void SetPageHeight(double height)
```

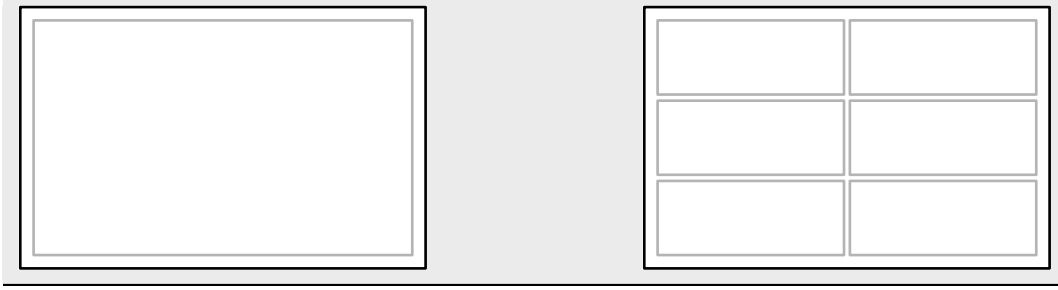
Sets the height of the pages of the report.

height
This number has to be greater than 0.0.

Example (*Figure: Example of using the SetPageHeight method*)

```
OEReportOptions opts = new OEReportOptions(3, 2);
opts.SetPageHeight(opts.GetPageHeight()*0.50);
```

Table 32: **Example of using the SetPageHeight method** (The scale of the shown pages is 1:4)



See also

- *OEReportOptions.SetPageOrientation* method
- *OEReportOptions.SetPageSize* method
- *OEReportOptions.GetPageWidth* method

SetPageMargin

```
void SetPageMargin(unsigned int margin, double size)
```

Sets the size of a specific page margin of the report.

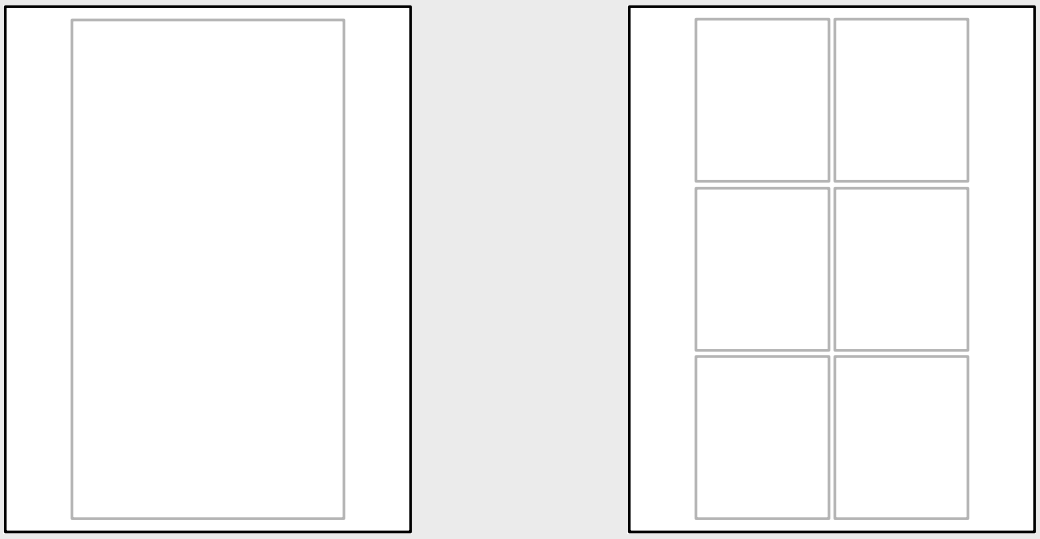
margin
: This value has to be from the *OEMargin* namespace.

size
: This value has to be a non-negative number.

Example (*Figure: Example of using the SetPageMargin method*)

```
OEReportOptions opts = new OEReportOptions(3, 2);
opts.SetPageMargin(OEMargin.Left,  100);
opts.SetPageMargin(OEMargin.Right, 100);
```

Table 33: **Example of using the SetPageMargin method** (The scale of the shown pages is 1:4)



See also

- *OEReportOptions.GetPageMargin* method
- *OEReportOptions.SetPageMargins* method

SetPageMargins

```
void SetPageMargins(double size)
```

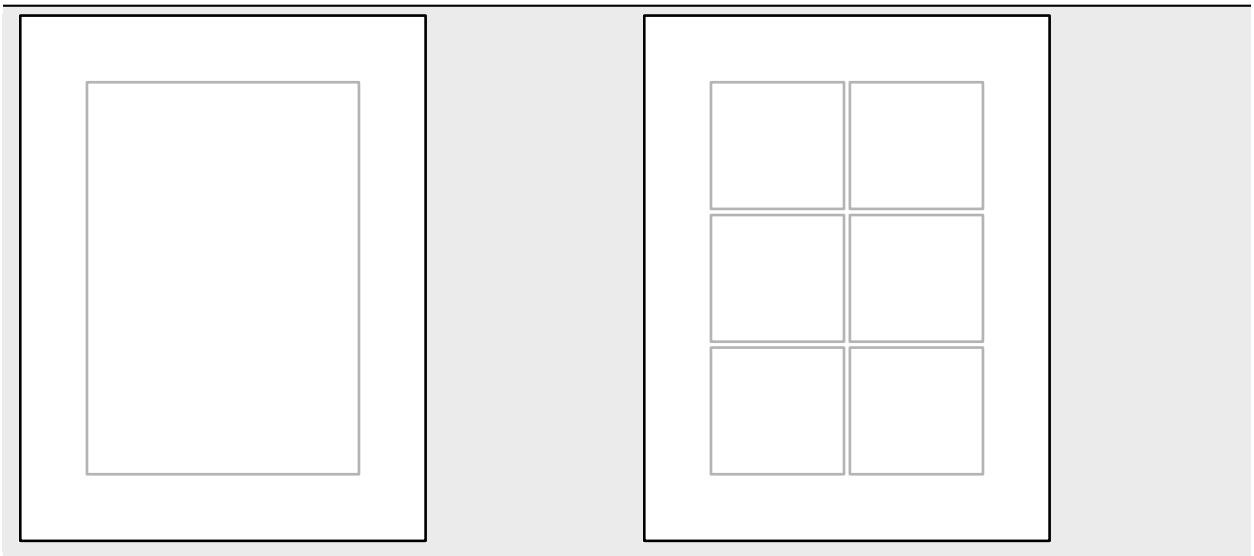
Sets the size of all the margins of the report.

size
: This value has to be a non-negative number.

Example (*Figure: Example of using the SetPageMargins method*)

```
OEReportOptions opts = new OEReportOptions(3, 2);
opts.SetPageMargins(100);
```

Table 34: **Example of using the SetPageMargins method** (The scale of the shown pages is 1:4)



See also

- *OEReportOptions.GetPageMargin* method
- *OEReportOptions.SetPageMargin* method

SetPageOrientation

```
void SetPageOrientation(unsigned int orientation)
```

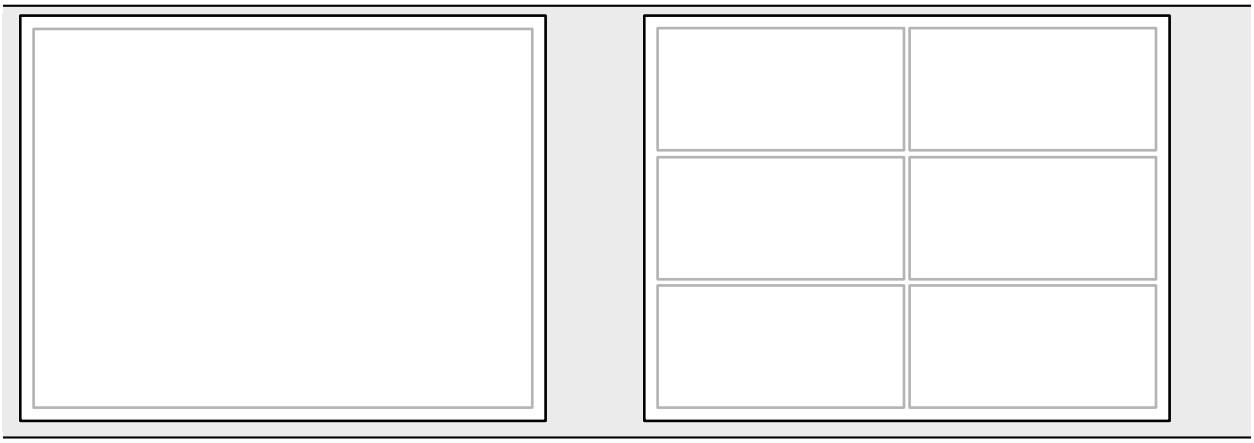
Sets the orientation of the pages of the report.

orientation
: This value has to be from the *OEPageOrientation* namespace.

Example (*Figure: Example of using the SetPageOrientation method*)

```
OEReportOptions opts = new OEReportOptions(3, 2);
opts.SetPageOrientation(OEPageOrientation.Landscape);
```

Table 35: **Example of using the SetPageOrientation method** (The scale of the shown pages is 1:4)



See also

- *OEReportOptions.SetPageHeight* method
- *OEReportOptions.SetPageSize* method
- *OEReportOptions.SetPageWidth* method

SetPageSize

```
void SetPageSize(unsigned int pagesize)
```

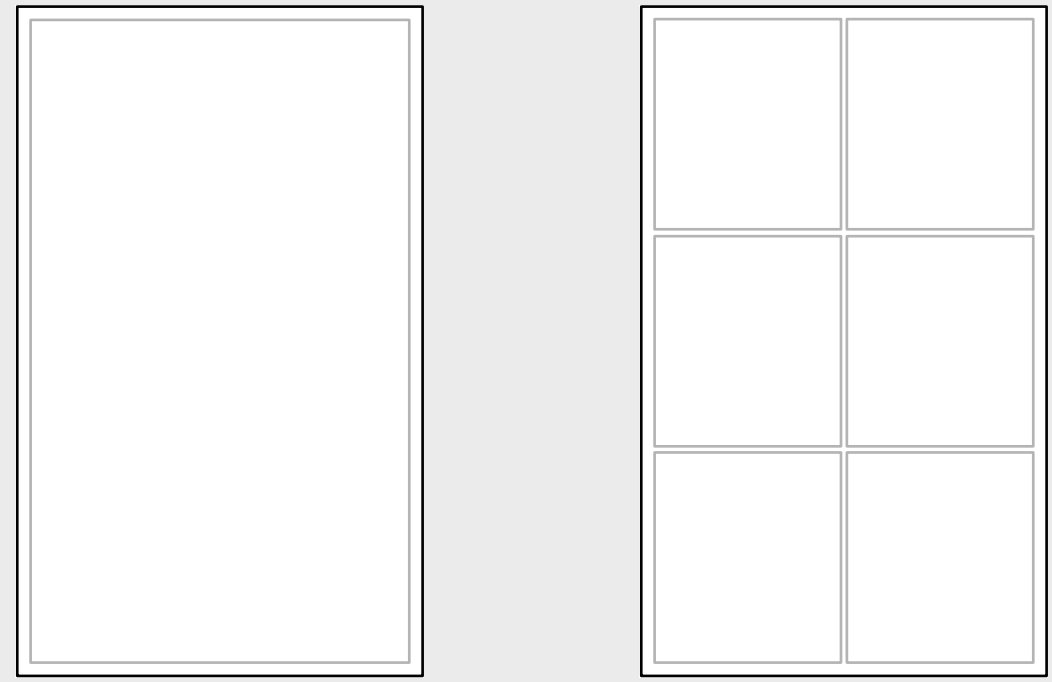
Sets the width and height of the pages of the report.

pagesize
This value has to be from the *OEPageSize* namespace.

Example (*Figure: Example of using the SetPageSize method*)

```
OEReportOptions opts = new OEReportOptions(3, 2);
opts.SetPageSize(OEPageSize.US_Legal);
```

Table 36: **Example of using the SetPageSize method** (The scale of the shown pages is 1:4)



See also

- *OEReportOptions.SetPageOrientation* method
- *OEReportOptions.SetPageHeight* method
- *OEReportOptions.SetPageWidth* method

SetPageWidth

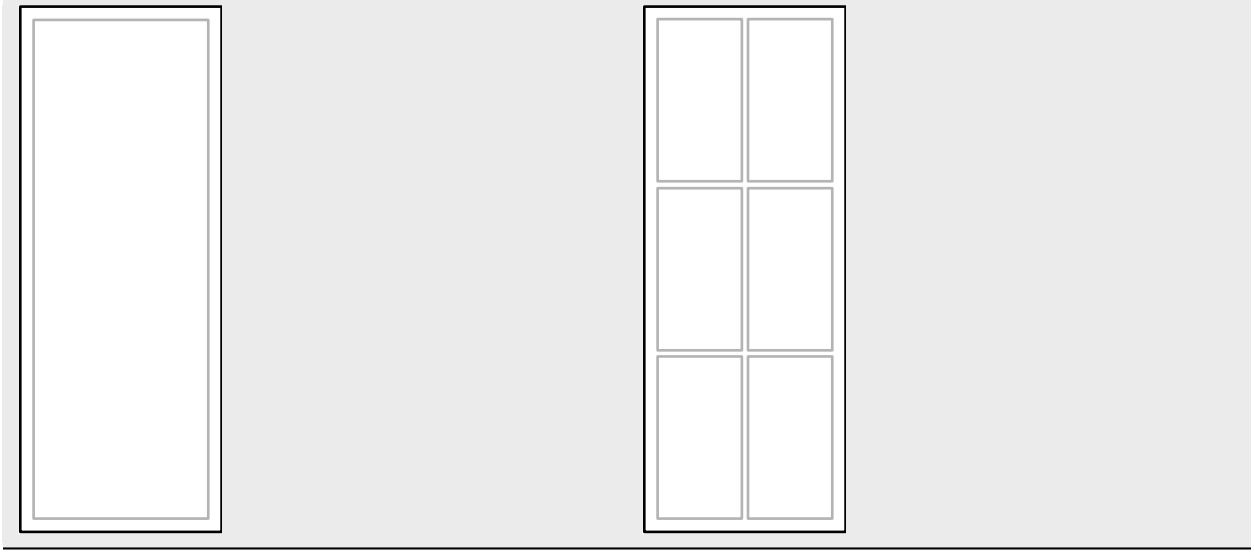
```
void SetPageWidth(double width)
```

Sets the width of the pages of the report.

width
: This number has to be greater than 0.0.

Example (*Figure: Example of using the SetPageWidth method*)

Table 37: **Example of using the SetPageWidth method** (The scale of the shown pages is 1:4)



See also

- *OEReportOptions.SetPageOrientation* method
- *OEReportOptions.SetPageHeight* method
- *OEReportOptions.SetPageSize* method

4.1.55 OESVGClass

```
class OESVGClass
```

SVG class is a name that can be assigned with SVG elements.

See also

- *OEIsValidSVGClassName* function
- *Generating Interactive SVG Images* chapter
- *OESVGGroup* class

Constructors

To create an *OESVGClass* object on an image use the *OEImageBase.NewSVGClass* method.

GetName

```
std::string GetName() const
```

Returns the name of the *OESVGClass* object.

4.1.56 OESVGGroup

```
class OESVGGroup
```

SVG group is a container for grouping and naming collections of SVG drawing elements.

See also

- *OEIsValidSVGGroupId* function
- *Generating Interactive SVG Images* chapter
- *OESVGClass* class

Constructors

To create an *OESVGGroup* object on an image use the *OEImageBase.NewSVGGroup* method.

AddSVGClass

```
bool AddSVGClass(const OESVGClass* svg_class)
```

Adds an *OESVGClass* object to the group. The SVG class and the SVG group have to be created on the same image.

GetId

```
std::string GetId() const
```

Returns the name of the *OESVGGroup* object.

IsVisible

```
bool IsVisible() const
```

Returns whether the objects drawn inside the given *OESVGGroup* will be visible when the image is opened.

4.2 OEDepict Constants

4.2.1 OE2DMolDisplaySetup

This namespace encodes symbolic constants used as bit-masks to indicate which molecule depiction parameters are being created when invoking the *OEConfigure2DMolDisplayOptions* function.

See also

- *OE2DMolDisplayOptions* class

All

The combination of following constants:

- *OE2DMolDisplaySetup.AromaticStyle*
- *OE2DMolDisplaySetup.AtomColorStyle*
- *OE2DMolDisplaySetup.AtomPropDisplay*
- *OE2DMolDisplaySetup.AtomStereoStyle*
- *OE2DMolDisplaySetup.BondColorStyle*
- *OE2DMolDisplaySetup.BondPropDisplay*
- *OE2DMolDisplaySetup.BondStereoStyle*
- *OE2DMolDisplaySetup.DefaultLineWidth*
- *OE2DMolDisplaySetup.HydrogenStyle*
- *OE2DMolDisplaySetup.ProtectiveGroupStyle*
- *OE2DMolDisplaySetup.Scale*
- *OE2DMolDisplaySetup.SuperAtomStyle*

AromaticStyle

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the aromatic ring style.

```
Contents of parameter -aromstyle
    Aliases : -astl
    Type : string
    Allow list : false
    Default : Kekule
    Simple : true
    Required : false
    Legal values : Kekule Circle Dash
    Brief : Aromatic ring display style
    Detail

        API : OE2DMolDisplayOptions::SetAromaticStyle method and OEAromaticStyle␣
↪namespace
```

See also

- *OEAromaticStyle* namespace
- *OE2DMolDisplayOptions.SetAromaticStyle* method

AtomColorStyle

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the atom color style.

```
Contents of parameter -atomcolor
    Aliases : -acolor
    Type : string
    Allow list : false
    Default : WhiteCPK
    Simple : true
    Required : false
    Legal values : WhiteCPK BlackCPK WhiteMonochrome BlackMonochrome cow cob bow wob
    Brief : Atom coloring style
    Detail

       API : OE2DMolDisplayOptions::SetAtomColorStyle method and OEAtomColorStyle␣
↪namespace

       cow          - (color on white) same as WhiteCPK
       cob          - (color on black) same as BlackCPK
       bow          - (black on white) same as WhiteMonochrome
       wob          - (white on black) same as BlackMonochrome
```

See also

- *OEAtomColorStyle* namespace
- *OE2DMolDisplayOptions.SetAtomColorStyle* method

AtomPropDisplay

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the atom property display.

```
Contents of parameter -atomprop
    Aliases : -aprop
    Type : string
    Allow list : false
    Default : Hidden
    Simple : true
    Required : false
    Legal values : Hidden AtomIdx MapIdx
    Brief : Atom property display
    Detail

        API : OE2DMolDisplayOptions::SetAtomPropertyFunctor method, OEDisplayNoAtomProp,␣
↪OEDisplayAtomIdx, and OEDisplayAtomMapIdx functors
```

See also

- *OEDisplayAtomIdx* class
- *OEDisplayAtomMapIdx* class
- *OEDisplayNoAtomProp* class
- *OE2DMolDisplayOptions.SetBondPropertyFunctor* method

AtomStereoStyle

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the atom stereo display style.

```
Contents of parameter -atomstereostyle

    Aliases : -atomsstl
    Type : string
    Allow list : false
    Default : AtomStereo|MDLStereoCenter|FancyStyle
    Simple : true
    Required : false
    Brief : Atom stereo display style
    Detail

        API : OE2DMolDisplayOptions::SetAtomStereoStyle method and OEAtomStereoStyle␣
↪namespace

        Primitive expressions :
        Hidden       - no atom stereo information is displayed
        AtomStereo   - display wedge/hash bonds
        CIPAtomStereo - display S/R CIP stereo labels
        MDLStereoCenter - display ABS/AND/OR stereo labels
        StandardStyle - standard hash/wedge style
        FancyStyle   - fancy hash/wedge style

        Combined expressions :
        Created by combining primitive tokens with |
```

See also

- *OEAtomStereoStyle* namespace
- *OE2DMolDisplayOptions.SetAtomStereoStyle* method

BondColorStyle

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the bond color style.

```
Contents of parameter -bondcolor
    Aliases : -bcolor
    Type : string
    Allow list : false
    Default : ByElement
    Simple : true
    Required : false
    Legal values : ByElement Monochrome
    Brief : Bond coloring style
    Detail

       API : OE2DMolDisplayOptions::SetBondColorStyle method and OEBondColorStyle␣
↪namespace
```

See also

- *OEBondColorStyle* namespace
- *OE2DMolDisplayOptions.SetBondColorStyle* method

BondPropDisplay

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the bond property display.

```
Contents of parameter -bondprop
    Aliases : -bprop
    Type : string
    Allow list : false
    Default : Hidden
    Simple : true
    Required : false
    Legal values : Hidden BondIdx
    Brief : Bond property display
    Detail

       API : OE2DMolDisplayOptions::SetBondPropertyFunctor method, OEDisplayNoBondProp␣
↪and OEDisplayBondIdx functors
```

See also

- *OEDisplayBondIdx* class
- *OEDisplayNoBondProp* class
- *OE2DMolDisplayOptions.SetBondPropertyFunctor* method

BondStereoStyle

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the bond stereo display style.

```
Contents of parameter -bondstereostyle
    Aliases : -bondsstl
    Type : string
    Allow list : false
    Default : BondStereo
    Simple : true
    Required : false
    Brief : Bond stereo display style
    Detail

        API : OE2DMolDisplayOptions::SetBondStereoStyle method and OEBondStereoStyle␣
↪namespace

        Primitive expressions :
        Hidden
        BondStereo
        CIPBondStereo

        Combined expressions :
        Created by combining primitive tokens with |
```

See also

- *OEBondStereoStyle* namespace
- *OE2DMolDisplayOptions.SetBondStereoStyle* method

DefaultLineWidth

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the default bond line width.

```
Contents of parameter -linewidth
    Aliases : -linew
    Type : int
    Allow list : false
```

(continues on next page)

(continued from previous page)

```
    Default : 2
    Simple : true
    Required : false
    Legal ranges :
        8 to 1
    Brief : Default bond line width
    Detail

       API : OE2DMolDisplayOptions::SetDefaultBondPen method and OEPen class
```

See also

- *OE2DMolDisplayOptions.SetDefaultBondPen* method
- *OEPen* class

HydrogenStyle

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the hydrogen display style.

```
Contents of parameter -hydrstyle
    Aliases : -hstl
    Type : string
    Allow list : false
    Default : ImplicitHetero|ExplicitAll
    Simple : true
    Required : false
    Brief : Hydrogen display style
    Detail

        API : OE2DMolDisplayOptions::SetHydrogenStyle method and OEHydrogenStyle␣
↪namespace

        Primitive expressions :
        Hidden
        ImplicitHetero
        ImplicitTerminal
        ImplicitAll
        ExplicitHetero
        ExplicitTerminal
        ExplicitAll

        Combined expressions :
        Created by combining primitive tokens with |
```

See also

- *OEHydrogenStyle* namespace

- *OE2DMolDisplayOptions.SetHydrogenStyle* method

ProtectiveGroupStyle

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the protective group display style.

```
Contents of parameter -protgroupdisp
    Aliases : -pgroup
    Type : string
    Allow list : false
    Default : Off
    Simple : true
    Required : false
    Brief : Protective group display style
    Detail

        API : OE2DMolDisplayOptions::SetProtectiveGroupStyle method and␣
↪OEProtectiveGroupStyle namespace

        Primitive expressions :
        Off             -  no protective group detection
        All             -  detect all protective groups
        Ac              -  R-C(=O)C
        Ad              -  R-C12CC3CC(C1)CC(C3)C2
        Alloc           -  R-C(=O)OCC=C
        Bn,Bzl,Benzyl   -  R-Cc1ccccc1
        BOC             -  R-C(=O)OC(C)(C)(C)
        Bom             -  R-COCc1ccccc1
        Bpoc            -  R-C(=O)OC(C)(C)c1ccc(cc1)c2ccccc2
        Bs              -  R-S(=O)(=O)c1ccc(cc1)Br
        Bt              -  R-n1c2ccccc2nn1
        Btm             -  R-CSc1ccccc1
        Bz, Benzoyl     -  R-C(=O)c1ccccc1
        Bzh             -  R-C(c1ccccc1)c2ccccc2
        BzOM            -  R-COc1ccccc1
        cHx             -  R-C1CCCCC1
        Dan, Dansyl     -  R-S(=O)(=O)c1cccc2c1cccc2N(C)C
        DCB             -  R-Cc1c(cccc1Cl)Cl
        DDE             -  R-C(=C1C(=O)CC(CC1=O)(C)C)C
        DDIV            -  R-C(=C1C(=O)CC(CC1=O)(C)C)CC(C)C
        DEIPS           -  R-[Si](CC)(CC)C(C)C
        DMAB            -  R-Cc1ccc(cc1)NC(=C2C(=O)CC(CC2=O)(C)C)CC(C)C
        DMIPS           -  R-[Si](C)(C)C(C)C
        DMPM            -  R-Cc1ccc(c(c1)OC)OC
        DMPS            -  R-[Si](C)(C)c1ccccc1
        DMTr            -  R-C(c1ccccc1)(c2ccc(cc2)OC)c3ccc(cc3)OC
        DNBZ            -  R-C(=O)c1cc(cc(c1)[N+](=O)[O-])[N+](=O)[O-]
        DNP             -  R-c1ccc(cc1[N+](=O)[O-])[N+](=O)[O-]
        DOC             -  R-C(=O)OC(C(C)C)C(C)C
        DPIPS           -  R-[Si](c1ccccc1)(c2ccccc2)OC(C)C
```

(continues on next page)

(continued from previous page)

```
        Dpp             -  R-P(=O)(c1ccccc1)c2ccccc2
        DTBMS           -  R-[Si](C)(C(C)(C)C)C(C)(C)C
        EE              -  R-C(C)OCC
        ESC             -  R-C(=O)OCCS(=O)(=O)CC
        Fmoc            -  R-C(=O)OCC3c1ccccc1c2ccccc23
        FourOMeBz       -  R-C(=O)c1ccc(cc1)OC
        FPMP            -  R-C1(CCN(CC1)c2ccccc2F)OC
        Im              -  R-n1ccnc1
        Lev             -  R-C(=O)CCC(=O)C
        MDIPS           -  R-[Si](C)(C(C)C)C(C)C
        MDPS            -  R-[Si](C)(c1ccccc1)c2ccccc2
        MEM             -  R-COCCOC
        Mes             -  R-c1c(cc(cc1C)C)C
        MMTR            -  R-C(c1ccccc1)(c2ccccc2)c3ccc(cc3)OC
        MOM             -  R-COC
        Moz             -  R-C(=O)OCc1ccc(cc1)OC
        MPC             -  R-C(=O)OC(C)(C)c1ccc(cc1)C
        MPM             -  R-c1ccccc1OC
        MS              -  R-S(=O)(=O)C
        MTHP            -  R-C1(CCOCC1)OC
        MTM             -  R-CSC
        MTr             -  R-S(=O)(=O)c1c(cc(c(c1C)C)OC)C
        NPE             -  R-CCc1ccc(cc1)[N+](=O)[O-]
        NPEOC           -  R-C(=O)OCCc1ccc(cc1)[N+](=O)[O-]
        NPES            -  R-S(=O)(=O)CCc1ccc(cc1)[N+](=O)[O-]
        Np              -  R-c1ccc(cc1)[N+](=O)[O-]
        NPS             -  R-Sc1ccccc1[N+](=O)[O-]
        NVOC            -  R-C(=O)OCc1cc(c(cc1[N+](=O)[O-])OC)OC
        OBzl            -  R-OCc1ccccc1
        ONOS            -  R-S(=O)(=O)c1ccccc1[N+](=O)[O-]
        OtBu            -  R-OC(C)(C)C
        OTf             -  R-S(=O)(=O)(C(F)(F)F)
        PAC             -  R-CC(=O)c1ccccc1
        PBF             -  R-S(=O)(=O)c1c(c(c2c(c1C)CC(O2)(C)C)C)C
        PCB             -  R-c1ccc(cc1)Cl
        PHF             -  R-C1(c2ccccc2-c3c1cccc3)c4ccccc4
        Pht             -  R-C(=O)c1ccccc1C=O
        Piv             -  R-C(=O)C(C)(C)C
        PMB             -  R-Cc1ccc(cc1)OC
        PMBM            -  R-COCc1ccc(cc1)OC
        Poc             -  R-C(=O)OC1CCCC1
        PPi             -  R-P(=O)([O-])OP(=O)([O-])[O-]
        SEM             -  R-COCC[Si](C)(C)C
        SES             -  R-S(=O)(=O)CC[Si](C)(C)C
        StBu            -  R-SC(C)(C)C
        TBDPS           -  R-[Si](C(C)(C)(C))(c1ccccc1)(c1ccccc1)
        TBMPS           -  R-[Si](c1ccccc1)(C(C)(C)C)C(C)(C)C
        TBS, TBDMS      -  R-[Si](C)(C)C(C)(C)C
        TCP             -  R-c1cc(c(cc1Cl)Cl)Cl
        Teoc            -  R-C(=O)OCC[Si](C)(C)C
        TES             -  R-[Si](CC)(CC)CC
        TFA             -  R-C(=O)C(F)(F)F
```

(continues on next page)

(continued from previous page)

```
        Thexyl          -  R-C(C)(C)C(C)C
        THF             -  R-C1CCCO1
        THP             -  R-C1CCCCO1
        TIPS            -  R-[Si](C(C)C)(C(C)C)C(C)C
        TMOB            -  R-Cc1c(cc(cc1OC)OC)OC
        TMS             -  R-[Si](C)(C)C
        transCinnamyl   -  R-C/C=C/c1ccccc1
        Trisyl          -  R-S(=O)(=O)c1c(cc(cc1C(C)C)C(C)C)C(C)C
        Troc            -  R-C(=O)OCC(Cl)(Cl)Cl
        TRT             -  R-C(c1ccccc1)(c2ccccc2)c3ccccc3
        TS, Tos         -  R-S(=O)(=O)c1ccc(cc1)C
        Xyl             -  R-c1c(cccc1C)C
        Z, Cbz          -  R-C(=O)OCc1ccccc1
```

Scale

```
Contents of parameter -scale
    Aliases : -s
    Type : float
    Allow list : false
    Default : 0.0
    Simple : true
    Required : false
    Legal ranges :
       500.0 to 0.0
    Brief :  Scaling of the depicted molecule
    Detail

        API : OE2DMolDisplayOptions::SetScale method

        0.0 value means that the molecule is scaled to
        auto-fit the (width/height) dimensions of the image
```

See also

- *OE2DMolDisplayOptions.SetScale* method
- *Controlling the Size of Depicted Molecules* section

SuperAtomStyle

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the super atom display style.

```
Contents of parameter -superdisp
Aliases : -super
Type : string
Allow list : false
```

(continues on next page)

(continued from previous page)

```
Default : Off
Simple : true
Required : false
Brief : Super atom display style
Detail

   API : OE2DMolDisplayOptions::SetSuperAtomStyle method and OESuperAtomStyle namespace

   Primitive expressions :
   Off          -  no super atom detection
   COOH         -  R-C(=O)OH
   COO_         -  R-C(=O)[O-]
   CHO          -  R-C(=O)H
   NC           -  R-C#[NX1]
   CN           -  R-[N+]#[C-]
   N2           -  R=[N+]=[N-]
   N3           -  R-N=[N+]=[N-]
   NCO          -  R-N=C=O
   NO2          -  R-N(=O)(=0) or R-[N+]([O-])(=0)
   CNO          -  R-C#[N+]-[O-]
   ONO          -  R-O-N=0
   SO3H         -  R-S(=O)(=O)(-OH)
   SO3_         -  R-S(=O)(=O)(-[O-])
   PO3H2        -  R-P(=O)(-OH)(-OH)
   CF3          -  R-C(-F)(-F)(-F)
   CCl3         -  R-C(-Cl)(-Cl)(-Cl)
   Me           -  R-CH3
   Et           -  R-CH2-CH3
   Pr           -  R-CH2-CH2-CH3
   Bu           -  R-CH2-CH2-CH2-CH3
   iBu          -  R-CH2-CH(-CH3)(-CH3)
   sBu          -  R-CH(-CH2-CH3)(-CH3)
   tBu          -  R-C(-CH3)(-CH3)(-CH3)
   OMe          -  R-O-CH3
   OEt          -  R-O-CH2-CH3
   OPr          -  R-O-CH2-CH2-CH3
   OBu          -  R-O-CH2-CH2-CH2-CH3
   All          -  Detect all super atoms

   Combined expressions :
   Created by combining primitive tokens with |
```

See also

- *OESuperAtomStyle* namespace
- *OE2DMolDisplayOptions.SetSuperAtomStyle* method

TitleLocation

Passing this constant to the *OEConfigure2DMolDisplayOptions* function result in generating the following default interface to configure the location of the molecule title.

```
Contents of parameter -titleloc
    Aliases : -tloc
    Type : string
    Allow list : false
    Default : Top
    Simple : true
    Required : false
    Legal values : Hidden Top Bottom
    Brief : Location of the molecule title
    Detail

       API : OE2DMolDisplayOptions::SetTitleLocation method and OETitleLocation namespace
```

See also

- *OETitleLocation* namespace
- *OE2DMolDisplayOptions.SetTitleLocation* method

4.2.2 OE2DPathPointType

This namespace contains constants representing types of points added to a *OE2DPath* object.

See also

- *OE2DPathPoint* class
- *OE2DPath.GetPoints* method

CurveC1

The first control point of a curve segment added to a path with the *OE2DPath.AddCurveSegment* method. See point marked as 'c1' in *Figure: Example of the points of a curve segment*.

CurveC2

The second control point of a curve segment added to a path with the *OE2DPath.AddCurveSegment* method. See point marked as 'c2' in *Figure: Example of the points of a curve segment*.

CurveEnd

The end point of a curve segment added to a path with the *OE2DPath.AddCurveSegment* method. See point marked as 'end' in *Figure: Example of the points of a curve segment*.

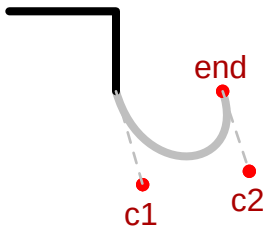


Fig. 212: **Example of the points of a curve segment**

LineEnd

Represents the end point of a line segment added to a path with the *OE2DPath.AddLineSegment* method. See point marked as 'end' in *Figure: Example of the points of a line segment*.

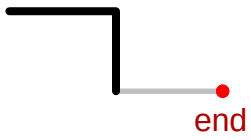


Fig. 213: **Example of the point of a line segment**

Start

Represents the first point of a path.

4.2.3 OEAlignment

This namespace contains constants that control how texts are aligned.

See also

- *OEFont.GetAlignment* method
- *OEFont.SetAlignment* method
- *OEImageBase.DrawText*

The *OEAlignment* namespace contains the following constants:

Default

The default text alignment style is *OEAlignment.Center*.

Center

The text is aligned to the center of the position passed as the first parameter of the *OEImageBase.DrawText* method. (This position is marked with a red dot in *Figure: Example of using the 'Center' alignment*)

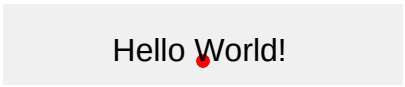


Fig. 214: **Example of using the 'Center' alignment**

Left

The text is left adjusted to the position passed as the first parameter of the *OEImageBase.DrawText* method. (This position is marked with a red dot in *Figure: Example of using the 'Left' alignment*)

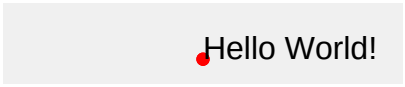


Fig. 215: **Example of using the 'Left' alignment**

Right

The text is right adjusted to the position passed as the first parameter of the *OEImageBase.DrawText* method. (This position is marked with a red dot in *Figure: Example of using the 'Right' alignment*)

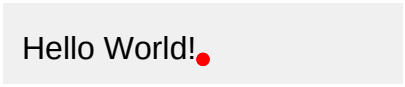
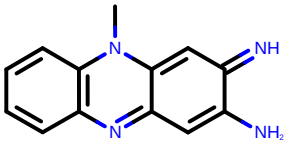
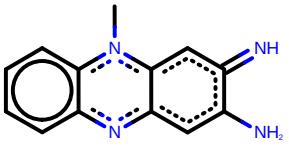
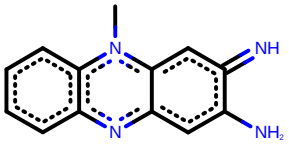
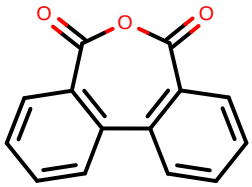
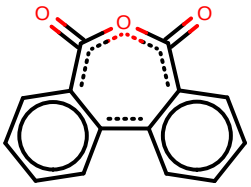
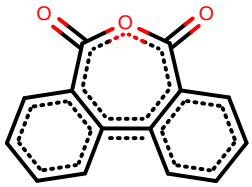
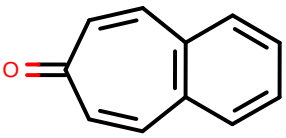
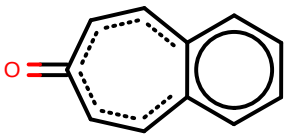
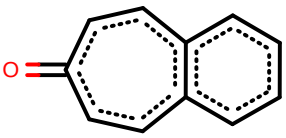
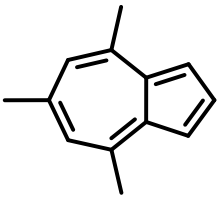
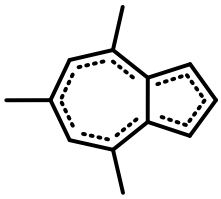
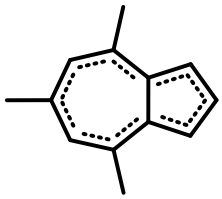


Fig. 216: **Example of using the 'Right' alignment**

4.2.4 OEAromaticStyle

This namespace contains constants representing various styles of aromatic ring depictions.

Table 38: **Examples of aromatic display styles**

	OEAromaticStyle.Kekule (default)	*OEAromaticStyle.Circle*	*OEAromaticStyle.Dash*
example A			
example B			
example C			
example D			

See also

- Aromaticity Perception chapter in **OEChem TK** manual
- *OE2DMolDisplayOptions.GetAromaticStyle*
- *OE2DMolDisplayOptions.SetAromaticStyle*

The *OEAromaticStyle* namespace contains the following constants:

Default

The default aromatic style is *OEAromaticStyle.Kekule*.

Circle

Aromatic rings up to 6-membered are depicted by drawing a circle inside. See example in *Figure: Example of using the 'Circle' aromaticity style*.

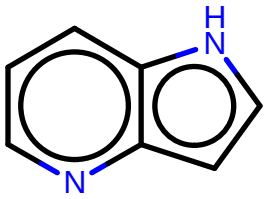


Fig. 217: **Examples of using the 'Circle' aromaticity style**

Note

OEDepict TK applies the "circle" style only for rings up to 6-membered. The remaining aromatic bonds are depicted using the "dash" style. See example B and C in *Table Examples of aromatic display styles*. **OEDepict TK** does not apply the "circle" style either if not all bonds of the rings are perceived as aromatic. See example A and D in *Table Examples of aromatic display styles*

Dash

See example in *Figure: Example of using the 'Dash' aromaticity style*.

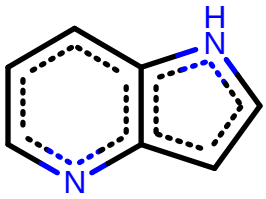


Fig. 218: **Example of using the 'Dash' aromaticity style**

Kekule

See example in *Figure: Example of using the 'Kekule' aromaticity style*.

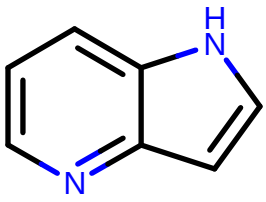


Fig. 219: **Example of using the 'Kekule' aromaticity style**

4.2.5 OEAtomColorStyle

This namespace contains constants representing various atom coloring styles of molecule depiction.

See also

- *OE2DMolDisplayOptions.GetAtomColorStyle* method
- *OE2DMolDisplayOptions.SetAtomColorStyle* method
- *OEGetDefaultAtomColor* function

After setting the atom color style of an *OE2DMolDisplayOptions* object:

- The color of the background can be changed by using the *OE2DMolDisplayOptions.SetBackgroundColor* method.
- The color that is associated with a specific atomic number can be changed by using the *OE2DMolDisplayOptions.SetAtomColor* method.

The *OEAtomColorStyle* namespace contains the following constants:

Default

The default atom color style is *OEAtomColorStyle.WhiteCPK*.

BlackCPK

The color of the background is *OEDarkBackgroundColor*. The atoms are colored using the CPK color convention. See example in *Figure: Example of using the 'BlackCPK' atom color style*.

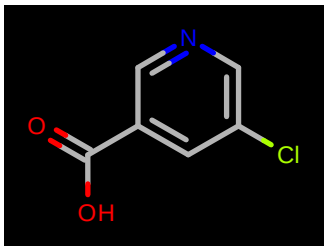


Fig. 220: **Example of using the 'BlackCPK' atom color style**

See also

- *Appendix: Element coloring (CPK)* chapter

BlackMonochrome

The color of the background is `OEDarkBackgroundColor`. Atoms and bonds are depicted using the `OEWhite` color. See example in *Figure: Example of using the 'BlackMonochrome' atom color style*.

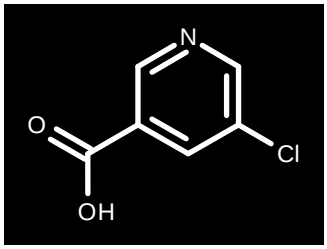


Fig. 221: **Example of using the 'BlackMonochrome' atom color style**

WhiteCPK

The color of the background is `OELightBackgroundColor`. The atoms are colored using the CPK color convention. See example in *Figure: Example of using the 'WhiteCPK' atom color style*.

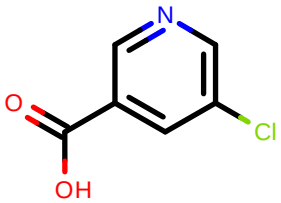


Fig. 222: **Example of using the 'WhiteCPK' atom color style**

See also

- *Appendix: Element coloring (CPK)* chapter

WhiteMonochrome

The color of the background is `OELightBackgroundColor`. Atoms and bonds are depicted using the `OEBlack` color. See example in *Figure: Example of using the WhiteMonochrome' atom color style*.

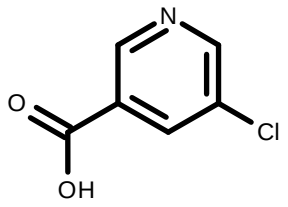


Fig. 223: **Example of using the 'WhiteMonochrome' atom color style**

4.2.6 OEAtomStereoStyle

This namespace encodes symbolic constants used as bit-masks to indicate what and how atom stereo information is displayed.

See also

- *OE2DMolDisplayOptions.GetAtomStereoStyle*
- *OE2DMolDisplayOptions.SetAtomStereoStyle*

The *OEAtomStereoStyle* namespace contains the following constants and nested namespaces:

Default

The combination of the following flags:

- *OEAtomStereoStyle.Display.Default*
- *OEAtomStereoStyle.HashWedgeStyle.Default*

See example in *Figure: Example of using the 'Default' atom stereo style*.

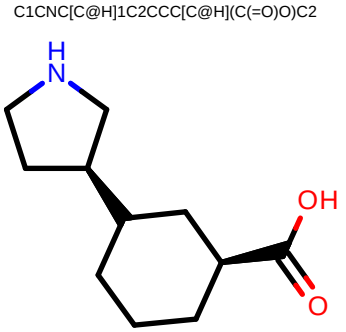


Fig. 224: **Example of using the 'Default' atom stereo style**

Hidden

Displays no atom stereo information. See example in *Figure: Example of using the 'Hidden' atom stereo style*.

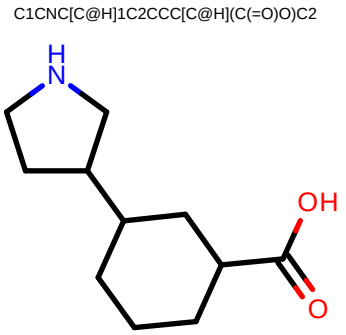


Fig. 225: **Example of using the 'Hidden' atom stereo style**

OEAtomStereoStyle::Display

This nested namespace encodes symbolic constants used as bit-masks to indicate what atom stereo information is displayed.

Default

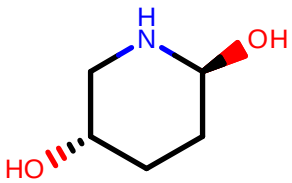
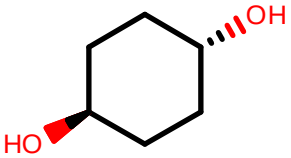
The combination of the following flags:

- *OEAtomStereoStyle.Display.AtomStereo*
- *OEAtomStereoStyle.Display.MDLStereoCenter*

AtomStereo

Displays atom stereo information in the form of hash/wedge bonds. See example in *Table: Example of using the 'AtomStereo' display style*.

Table 39: **Example of using the 'AtomStereo' display style**

O[C@@H]1CN[C@@H](O)CC1	*O[C@@H]1CC[C@@H](O)CC1*
	

If the molecule is imported from an MOL or SDF file with an "either" stereo than it is depicted as a wavy bond. See example in *Figure: Example of depiction of up, down and either (wavy) bonds*.

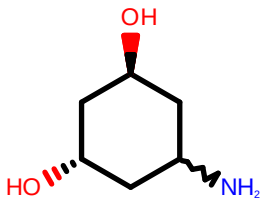
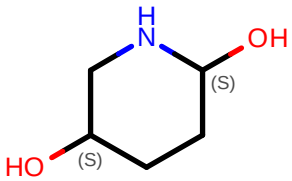
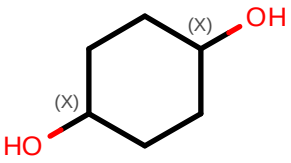


Fig. 226: **Example of depiction of up, down and either (wavy) bonds**

CIPAtomStereo

Displays the CIP atom stereo information next to the corresponding atom. See example in *Table: Example of using the 'CIPAtomStereo' display style*.

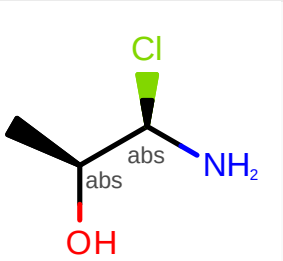
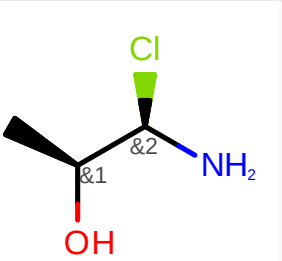
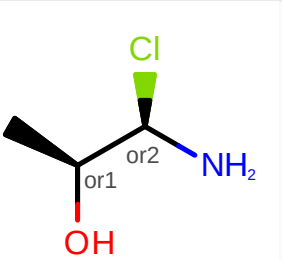
Table 40: **Example of using the 'CIPAtomStereo' display style**

O[C@@H]1CN[C@@H](O)CC1	*O[C@@H]1CC[C@@H](O)CC1*
	

MDLStereoCenter

Displays the MDL enhanced stereochemical atom labels.

Table 41: **Example of using the 'MDLStereoCenter' display style**

MDL stereo group type	Absolute	AND	OR
Label in molfile	STEABS	STERACn	STERELn
Display label	'abs'	'&n'	'orn'
Example			

All

The combination of the following flags:

- *OEAtomStereoStyle.Display.AtomStereo*
- *OEAtomStereoStyle.Display.CIPAtomStereo*
- *OEAtomStereoStyle.Display.MDLStereoCenter*

OEAtomStereoStyle::HashWedgeStyle

This nested namespace encodes symbolic constants used to indicate the style of the hash/wedge bonds that depict atom stereo information.

Default

The default hash/wedge display style is *OEAtomStereoStyle.HashWedgeStyle.Fancy*

Fancy

See example in *Figure: Example of using the 'Fancy' hash/wedge style*.

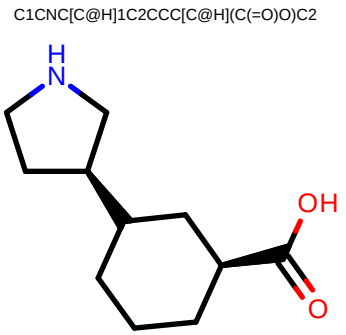


Fig. 227: **Example of using the 'Fancy' hash/wedge style**

Standard

See example in *Figure: Example of using the 'Standard' hash/wedge style*.

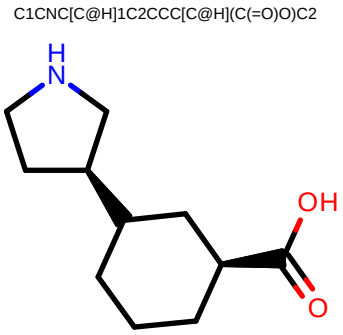


Fig. 228: **Example of using the 'Standard' hash/wedge style**

4.2.7 OEBondColorStyle

This namespace contains constants representing various bond coloring styles of molecule depiction.

See also

- *OE2DMolDisplayOptions.GetBondColorStyle*
- *OE2DMolDisplayOptions.SetBondColorStyle*

The *OEBondColorStyle* namespace contains the following constants:

Default

The default bond color style is *OEBondColorStyle.ByElement*

ByElement

See example in *Figure: Example of using the 'ByElement' bond color style*.

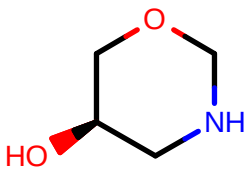


Fig. 229: **Example of using the 'ByElement' bond color style**

Monochrome

See example in *Figure: Example of using the 'Monochrome' bond color style*.

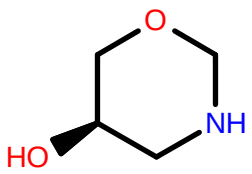


Fig. 230: **Example of using the 'Monochrome' bond color style**

4.2.8 OEBondDisplayPosition

This namespace controls how multi-line bonds are rendered. The *OEBondDisplayPosition* only affects bonds of the following types (as returned by *OE2DBondDisplay.GetDisplayType*):

- *OEBondDisplayType.Double*
- *OEBondDisplayType.AromaticDash*
- *OEBondDisplayType.SingleDouble*
- *OEBondDisplayType.SingleArom*
- *OEBondDisplayType.DoubleArom*
- *OEBondDisplayType.Aromatic*

The concept of *Left* and *Right* are relative to the "begin" and "end" atoms of the associated OEBondBase. The side of *Left* and *Right* is defined by standing on the "begin" atom and looking towards the "end" atom.

The following figure shows how both *Left* and *Right* double bonds can appear in a benzene ring.

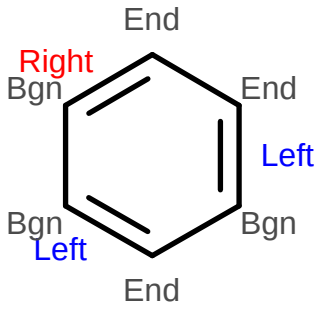


Fig. 231: *Example of various multi-line bond positions*

Note

The *OE2DBondDisplay* does not store pointers to the "begin" and "end" atoms. The "begin" and "end" atoms can be retrieved by accessing the associated OEBondBase through the *OEBondDisplayBase.GetBond* method and then calling the `OEBondBase.GetBgn` and `OEBondBase.GetEnd` methods respectively.

See also

- *OE2DBondDisplay.GetDisplayPos* method
- *OE2DBondDisplay.SetDisplayPos* method

The *OEBondDisplayPosition* namespace contains the following constants:

Center

Renders a double bond so that the "begin" and "end" atom coordinates end up between each line of the double bond. See example in *Figure: Example of a double bond rendered in the center*.

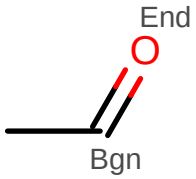


Fig. 232: *Example of a double bond rendered in the center*

Left

Renders a double bond by drawing a line between the "begin" and "end" atom coordinates, and then drawing a parallel line to the left of that line. See example in *Figure: Example of a double bond rendered with its parallel line to the left*.

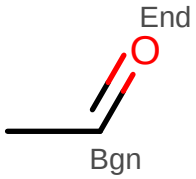


Fig. 233: *Example of a double bond rendered with its parallel line to the left*

Right

Renders a double bond by drawing a line between the "begin" and "end" atom coordinates, and then drawing a parallel line to the right of that line. See example in *Figure: Example of a double bond rendered with its parallel line to the right*.

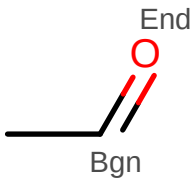


Fig. 234: *Example of a double bond rendered with its parallel line to the right*

4.2.9 OEBondDisplayType

This namespace contains constants representing various styles of bond representations.

See also

- `OE2DBondDisplay.GetDisplayType` method
- `OE2DBondDisplay.SetDisplayType` method

The `OEBondDisplayType` namespace contains the following constants:

Single

The `OEBondDisplayType.Single` type is used to represent single bonds. See example in *Figure: Example of depicting a 'Single' bond type*.



Fig. 235: **Example of depicting a 'Single' bond type**

Double

The `OEBondDisplayType.Double` type is used to represent double bonds. See example in *Figure: Example of depicting a 'Double' bond type*.



Fig. 236: **Example of depicting a 'Double' bond type**

Triple

The *OEBondDisplayType.Triple* type is used to represent triple bonds. See example in *Figure: Example of depicting a 'Triple' bond type*.



Fig. 237: **Example of depicting a 'Triple' bond type**

Quadruple

The *OEBondDisplayType.Quadruple* type is used to represent quadruple bonds. See example in *Figure: Example of depicting a 'Quadruple' bond type*.



Fig. 238: **Example of depicting a 'Quadruple' bond type**

Wedge

See example in *Figure: Example of depicting a 'Wedge' bond type*.



Fig. 239: **Example of depicting a 'Wedge' bond type**

Hash

See example in *Figure: Example of depicting a 'Hash' bond type*.



Fig. 240: **Example of depicting a 'Hash' bond type**

Wavy

See example in *Figure: Example of depicting a 'Wavy' bond type*.



Fig. 241: **Example of depicting a 'Wavy' bond type**

AromaticDash

The `OEBondDisplayType.AromaticDash` type is used to represent aromatic rings when the `OEAromaticStyle.Dash` style is turned on. See example in *Figure: Example of depicting an 'AromaticDash' bond type*.

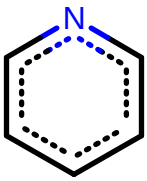


Fig. 242: **Example of depicting an 'AromaticDash' bond type**

DoubleBowTie

See example in *Figure: Example of depicting an 'DoubleBowTie' bond type*.



Fig. 243: **Example of depicting a 'DoubleBowTie' bond type**

SingleDouble

The `OEBondDisplayType.SingleDouble` type is used to represent MDL query bonds that can be matched to any single or double bonds in the target molecule(s) (*i.e.* bond type *5* in the MDL query files) See example in *Figure: Example of depicting an 'SingleDouble' bond type*.

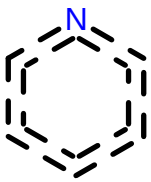


Fig. 244: **Example of depicting a 'SingleDouble' bond type**

SingleArom

The *OEBondDisplayType.SingleArom* type is used to represent MDL query bonds that can be matched to any single or aromatic bonds in the target molecule(s) (*i.e.* bond type *6* in the MDL query files) See example in *Figure: Example of depicting an 'SingleDouble' bond type*.

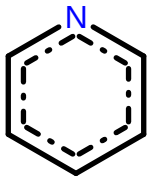


Fig. 245: **Example of depicting a 'SingleArom' bond type**

DoubleArom

The *OEBondDisplayType.DoubleArom* type is used to represent MDL query bonds that can be matched to any double or aromatic bonds in the target molecule(s) (*i.e.* bond type *7* in the MDL query files) See example in *Figure: Example of depicting an 'DoubleArom' bond type*.

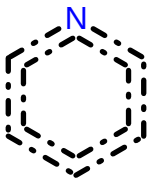


Fig. 246: **Example of depicting a 'DoubleArom' bond type**

Aromatic

The *OEBondDisplayType.Aromatic* type is used to represent MDL query bonds that can be matched to any aromatic bonds in the target molecule(s) (*i.e.* bond type *4* in the MDL query files) See example in *Figure: Example of depicting an 'Aromatic' bond type*.

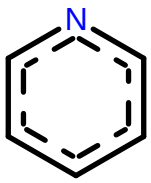


Fig. 247: **Example of depicting an 'Aromatic' bond type**

Any

The *OEBondDisplayType.Any* type is used to represent MDL query bonds that can be matched to any bonds in the target molecule(s) (*i.e.* bond type *8* in the MDL query files) See example in *Figure: Example of depicting an 'Any' bond type*.

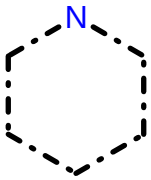


Fig. 248: **Example of depicting an 'Any' bond type**

Hidden

The bond is not displayed.

4.2.10 OEBondStereoStyle

This namespace encodes symbolic constants used as bit-masks to indicate what bond stereo information is displayed.

See also

- *OE2DMolDisplayOptions.GetBondStereoStyle*
- *OE2DMolDisplayOptions.SetBondStereoStyle*

The *OEBondStereoStyle* namespace contains the following constants and nested namespaces:

Default

The default bond stereo style is *OEBondStereoStyle.Display.Default*. See example in *Figure: Example of using the 'Default' bond stereo style*.

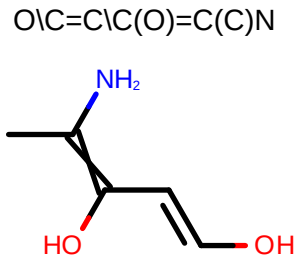


Fig. 249: **Example of using the 'Default' bond stereo style**

Hidden

Displays no bond stereo information. See example in *Figure: Example of using the 'Hidden' bond stereo style*.

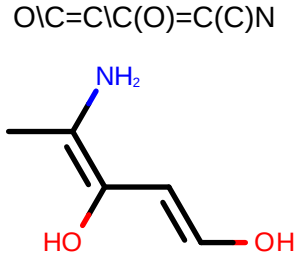


Fig. 250: **Example of using the 'Hidden' bond stereo style**

OEBondStereoStyle::Display

This nested namespace encodes symbolic constants used as bit-masks to indicate what bond stereo information is displayed.

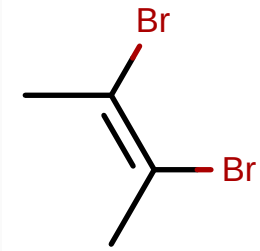
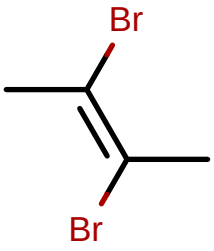
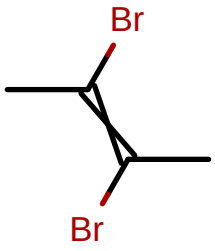
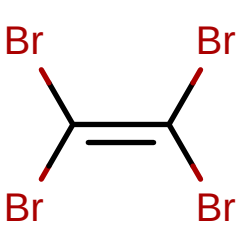
Default

The default bond stereo display style is `OEBondStereoStyle.Display.BondStereo`.

BondStereo

See example in *Table: Example of using the 'BondStereo' display style*.

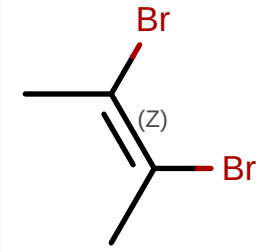
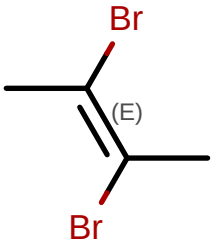
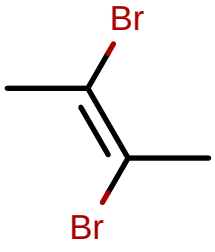
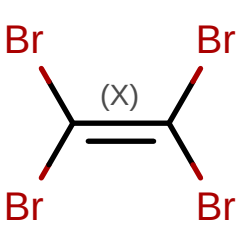
Table 42: **Example of using the 'BondStereo' display style**

cis	trans	undefined	over-defined
C\C(Br)=C(/C)Br	*C\C(Br)=C(\C)Br*	*CC(Br)=C(C)Br*	*Br\C(Br)=C(\Br)Br*
			

CIPBondStereo

Displays the CIP bond stereo information next to the corresponding bond. See example in *Table: Example of using the 'CIPBondStereo' display style*.

Table 43: **Example of using the 'CIPBondStereo' display style**

cis	trans	undefined	over-defined
C\C(Br)=C(/C)Br	*C\C(Br)=C(\C)Br*	*CC(Br)=C(C)Br*	*Br\C(Br)=C(\Br)Br*
			

All

The combination of the following flags:

- `OEBondStereoStyle.Display.CIPBondStereo`
- `OEBondStereoStyle.Display.BondStereo`

See example in *Table: Example of using the 'All' display style*.

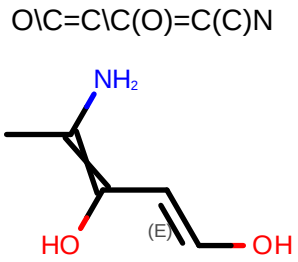


Fig. 251: **Example of using the 'All' display style**

4.2.11 OEDepict

This namespace contains constants.

OEDarkBackgroundColor

This color is used as a background color for the *OEAtomColorStyle.BlackMonochrome* and the *OEAtomColorStyle.BlackCPK* atom color styles.

OELightBackgroundColor

This color is used as a background color for the *OEAtomColorStyle.WhiteMonochrome* and the *OEAtomColorStyle.WhiteCPK* atom color styles.

OEBlackBoxPen

See example in *Figure: Example of using the 'OEBlackBoxPen' pen to draw a rectangle*.



Fig. 252: **Example of using the 'OEBlackBoxPen' pen to draw a rectangle**

OEBlackPen

See example in *Figure: Example of using the 'OEBlackPen' pen to draw a rectangle*.

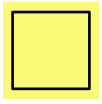


Fig. 253: **Example of using the 'OEBlackPen' pen to draw a rectangle**

OEBlueBoxPen

See example in *Figure: Example of using the 'OEBlueBoxPen' pen to draw a rectangle*.



Fig. 254: **Example of using the 'OEBlueBoxPen' pen to draw a rectangle**

OEBluePen

See example in *Figure: Example of using the 'OEBluePen' pen to draw a rectangle*.

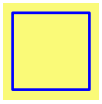


Fig. 255: **Example of using the 'OEBluePen' pen to draw a rectangle**

OEDefaultFont

OEGreenBoxPen

See example in *Figure: Example of using the 'OEGreenBoxPen' pen to draw a rectangle*.



Fig. 256: **Example of using the 'OEGreenBoxPen' pen to draw a rectangle**

OEGreenPen

See example in *Figure: Example of using the 'OEGreenPen' pen to draw a rectangle*.

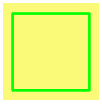


Fig. 257: **Example of using the 'OEGreenPen' pen to draw a rectangle**

OELightGreyBoxPen

See example in *Figure: Example of using the 'OELightGreyBoxPen' pen to draw a rectangle*.



Fig. 258: **Example of using the 'OELightGreyBoxPen' pen to draw a rectangle**

OELightGreyPen

See example in *Figure: Example of using the 'OELightGreyPen' pen to draw a rectangle*.

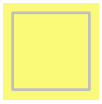


Fig. 259: **Example of using the 'OELightGreyPen' pen to draw a rectangle**

OERedBoxPen

See example in *Figure: Example of using the 'OERedBoxPen' pen to draw a rectangle*.



Fig. 260: **Example of using the 'OERedBoxPen' pen to draw a rectangle**

OERedPen

See example in *Figure: Example of using the 'OERedPen' pen to draw a rectangle*.

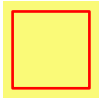


Fig. 261: **Example of using the 'OERedPen' pen to draw a rectangle**

OETransparentPen

Pen with fully transparent foreground and background colors.

OESVGAreaPen

Pen with almost transparent foreground and background colors.

OEWhiteBoxPen

See example in *Figure: Example of using the 'OEWhiteBoxPen' pen to draw a rectangle*.

Fig. 262: **Example of using the 'OEWhiteBoxPen' pen to draw a rectangle**

OEWhitePen

See example in *Figure: Example of using the 'OEWhitePen' pen to draw a rectangle*.

Fig. 263: **Example of using the 'OEWhitePen' pen to draw a rectangle**

4.2.12 OEDepictOrientation

This namespace contains constants that control the preferred orientation of the 2D coordinates when calling the *OEPrepareDepiction* function.

See also

- *OEPrepareDepictionOptions.GetDepictOrientation* method
- *OEPrepareDepictionOptions.SetDepictOrientation* method
- *OEPrepareDepiction* function

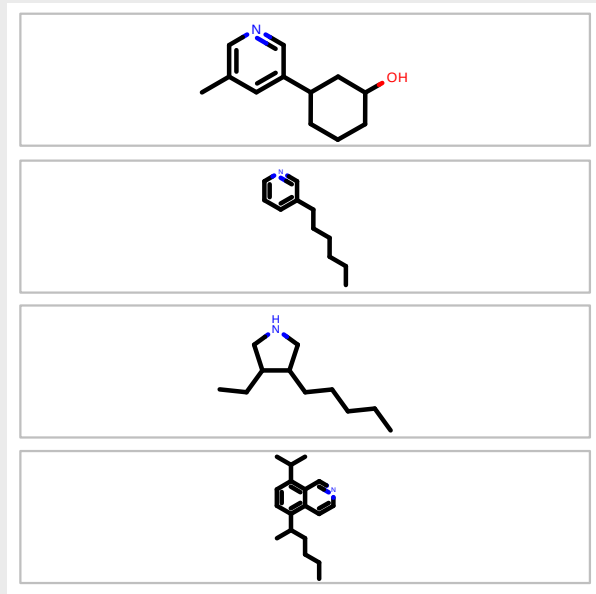
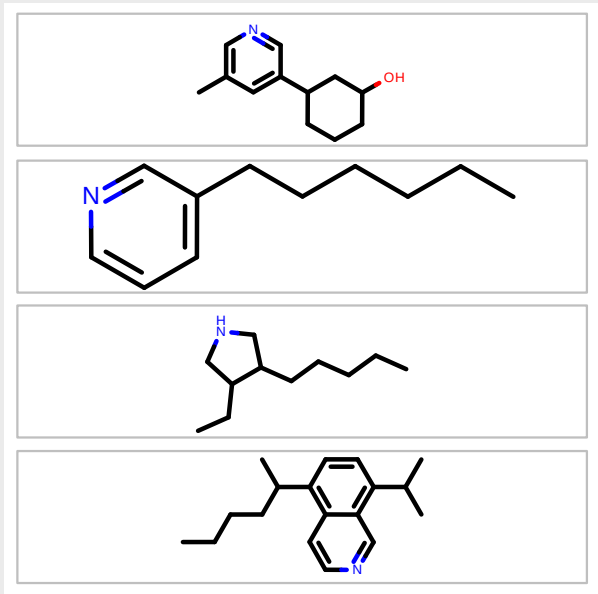
The *OEDepictOrientation* namespace contains the following constants:

Default

The default orientation is *OEDepictOrientation.Original*.

Horizontal

The orientation of the 2D coordinates is changed in order to maximize the scaling of the molecule for horizontal images. See example above of depicting molecules with *OEDepictOrientation.Default* and *OEDepictOrientation.Horizontal* orientations.

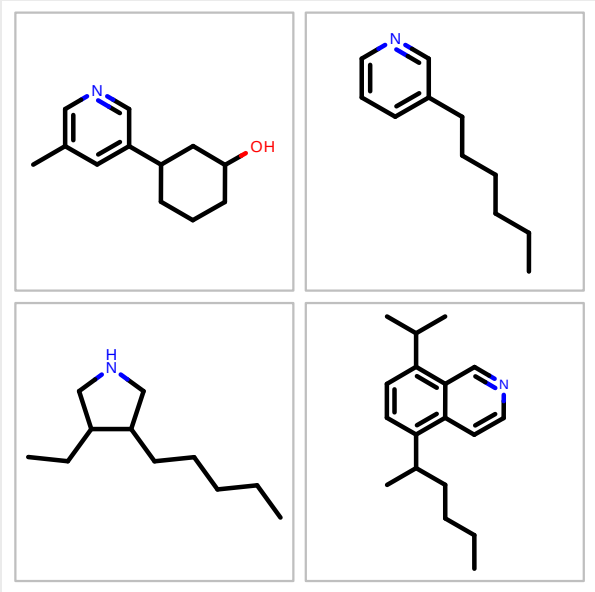
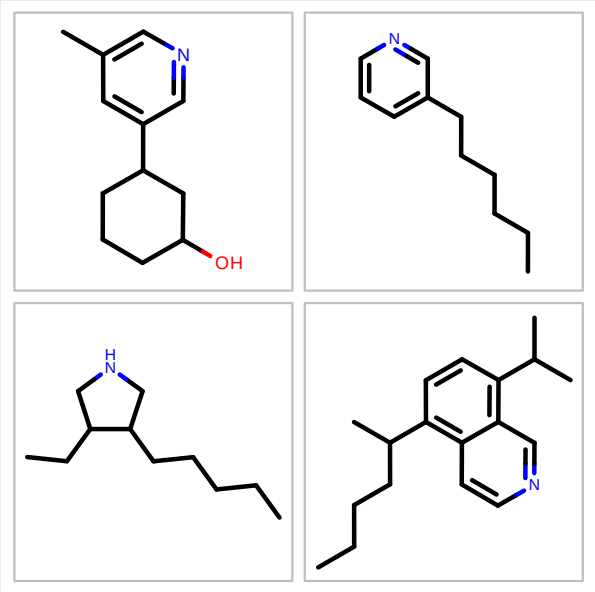
OEDepictOrientation.Default	*OEDepictOrientation.Horizontal*
	

Original

No-op *i.e* the orientation of the 2D coordinates are not optimized in order to maximize the molecule scaling.

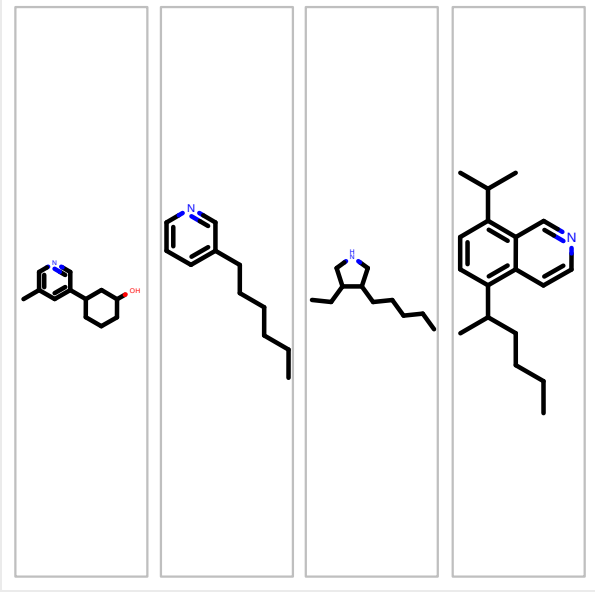
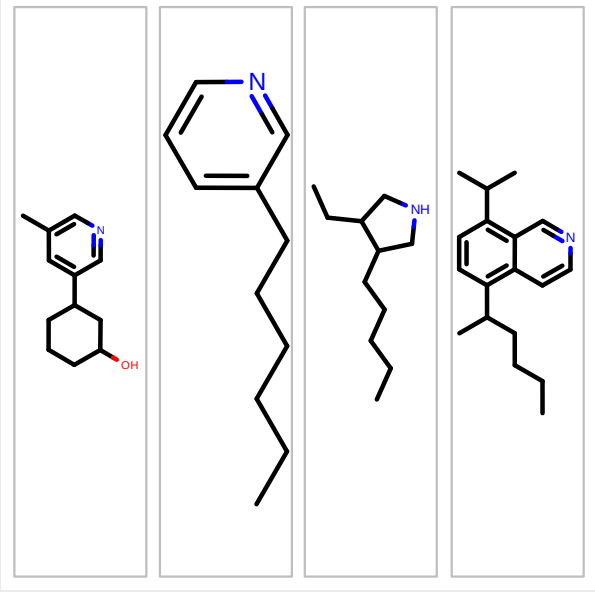
Square

The orientation of the 2D coordinates is changed in order to maximize the scaling of the molecule for square images. See example above of depicting molecules with *OEDepictOrientation.Default* and *OEDepictOrientation.Square* orientations.

OEDepictOrientation.Default	*OEDepictOrientation.Square*
	

Vertical

The orientation of the 2D coordinates is changed in order to maximize the scaling of the molecule for vertical images. See example above of depicting molecules with *OEDepictOrientation.Default* and *OEDepictOrientation.Vertical* orientations.

OEDepictOrientation.Default	*OEDepictOrientation.Vertical*
	

4.2.13 OEFill

This namespace contains constants representing whether or not background color of an *OEPen* object is used to *fill* a shape that is being drawn.

See also

- `OEPen.GetFill` method
- `OEPen.SetFill` method

The `OEFill` namespace contains the following constants:

Default

The default fill style is `OEFill.Off`.

Off

If the fill property of an *OEPen* object is **off** than the interior of the shape being drawn will be unfilled. See example in *Figure: Example of turning the object filling off*.

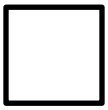


Fig. 264: **Example of turning the object filling off**

On

If the fill property of an *OEPen* object is **on** than its background color is used to fill the interior of the shape being drawn. See example in *Figure: Example of turning the object filling off*.



Fig. 265: **Example of turning the object filling on**

4.2.14 OEFontFamily

This namespace contains constants representing various font family styles of the *OEFont* class.

See also

- *OEFont.GetFamily* method
- *OEFont.SetFamily* method

The *OEFontFamily* namespace contains the following constants:

Default

The default font family is *OEFontFamily.Arial*.

Arial

Sans-serif typeface. See example in *Figure: Example of drawing text with 'Arial' font family*.

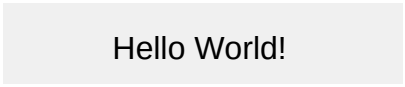


Fig. 266: **Example of drawing text with 'Arial' font family**

Courier

Mono-spaced slab serif typeface See example in *Figure: Example of drawing text with 'Courier' font family*.

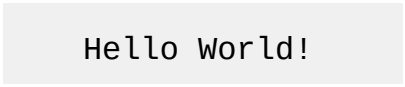


Fig. 267: **Example of drawing text with 'Courier' font family**

Helvetica

Sans-serif typeface. See example in *Figure: Example of drawing text with 'Helvetica' font family*.

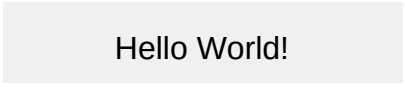


Fig. 268: **Example of drawing text with 'Helvetica' font family**

Times

Serif typeface. See example in *Figure: Example of drawing text with 'Times' font family*.



Fig. 269: **Example of drawing text with 'Times' font family**

4.2.15 OEFontStyle

This namespace encodes symbolic constants used as bit-masks to indicate the style of an *OEFont* object.

See also

- *OEFont.GetStyle* method
- *OEFont.SetStyle* method

The *OEFontStyle* namespace contains the following constants:

Default

The default font style is *OEFontStyle.Normal*.

Bold

See example in *Figure: Example of drawing text with 'Bold' font style*.



Fig. 270: **Example of drawing text with 'Bold' font style**

Italic

See example in *Figure: Example of drawing text with 'Italic' font style*.



Fig. 271: **Example of drawing text with 'Italic' font style**

Normal

See example in *Figure: Example of drawing text with 'Normal' font style*.

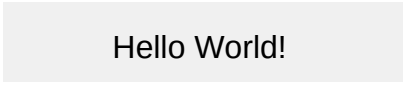


Fig. 272: **Example of drawing text with 'Normal' font style**

4.2.16 OEHighlightOverlayStyle

This namespace contains constants representing various styles of overlay highlighting.

See also

- *OEAddHighlightOverlay* function
- *OEHighlightOverlayBase* class
- *Highlighting Overlapped Patterns* section

The *OEHighlightOverlayStyle* namespace contains the following constants:

Default

The default highlighting style is *OEHighlightOverlayStyle.BallAndStick*.

BallAndStick

See example in *Figure: Example of overlay highlighting using the 'BallAndStick' style*.

- Atoms are highlighted by depicting spheres underneath them. If an atom is part of more than one patterns, then all the colors associated with the patterns are used by turns to color pie segments of the sphere.
- Bonds are highlighted by depicting sticks underneath them. If a bond is part of more than one patterns, then all the colors associated with the patterns are used by turns to color line segments of the stick.

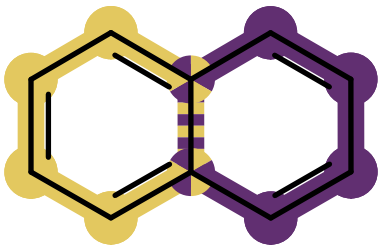


Fig. 273: **Example of overlay highlighting using the 'BallAndStick' style**

The radius of the spheres and the width of the sticks are automatically scaled with the molecule. See example in *Figure: Example of scaling the 'BallAndStick' style highlighting along with the molecule*.

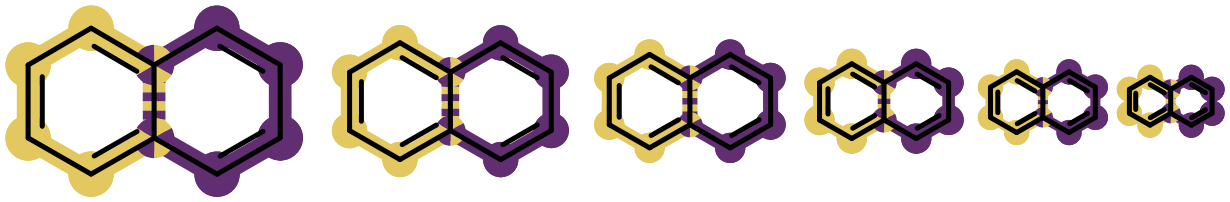


Fig. 274: **Example of scaling the 'BallAndStick' style highlighting along with the molecule**

See also

- *OEHighlightOverlayByBallAndStick* class

Note

It is recommended to select colors with high contrast when highlighting overlapped patterns with the `OEHighlightOverlayStyle.BallAndStick` style.

See also

- `OEGetContrastColors` function

Even though there is no limit on the number of overlapping patterns that can be highlighted simultaneously, attempting to highlight too many patterns will result in a complex image that will be difficult to visually interpret. See example in *Figure: Example of highlighting extremely overlapping patterns* In this image, all unique 'phenol' substructures of a molecule are highlighted simultaneously.

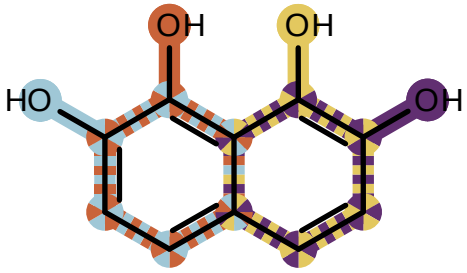


Fig. 275: **Example of highlighting extremely overlapping patterns**

4.2.17 OEHighlightParamName

This namespace contains the default parameter names used when the interface generated by invoking the *OEConfigureHighlightParams* function.

See also

- *OEHighlightSetup* namespace.

The *OEHighlightParamName* namespace contains the following constants:

Color

The default name of highlight color parameter is **"-highlightcolor"**.

See also

- *OEHighlightSetup.Color* constant
- *OEConfigureHighlightColor* function
- *OEGetHighlightColor* function

Style

The default name of highlight stylecolor parameter is **"-highlightstyle"**.

See also

- *OEHighlightSetup.Style* constant
- *OEConfigureHighlightStyle* function
- *OEGetHighlightStyle* function

4.2.18 OEHighlightSetup

This namespace encodes symbolic constants used as bit-masks to indicate which highlighting interface parameters are being created when invoking the *OEConfigureHighlightParams* function.

See also

- *OEHighlightParamName* namespace

The *OEHighlightSetup* namespace contains the following constants:

All

The combination of following constants:

- *OEHighlightSetup.Color*
- *OEHighlightSetup.Style*

Color

Passing this constant to the *OEConfigureHighlightParams* function result in generating the following default interface to configure the color of the highlighting by invoking the *OEConfigureHighlightColor* function.

```
Contents of parameter -highlightcolor
    Aliases : -hcolor
    Type : string
    Allow list : false
    Default : red
    Simple : true
    Required : false
    Legal values : black blue bluetint brass brown copper cyan darkblue
      darkbrown darkcyan darkgreen darkgrey darkmagenta darkorange
      darkpurple darkred darkrose darksalmon darkyellow gold green
      greenblue greentint grey hotpink lightblue lightbrown lightgreen
      lightgrey lightorange lightpurple lightsalmon limegreen magenta
      mediumblue mediumbrown mediumgreen mediumorange mediumpurple
      mediumsalmon mediumyellow olivebrown olivegreen olivegrey pewter
      orange pink pinktint purple red redorange royalblue seagreen silver
      skyblue violet white yellow yellowtint
```

See also

- *OEHighlightParamName.Color* constant
- *OEConfigureHighlightColor* function
- *OEGetHighlightColor* function

Style

Passing this constant to the *OEConfigureHighlightParams* function result in generating the following default interface to configure the style of the highlighting by invoking the *OEConfigureHighlightStyle* function.

```
Contents of parameter -highlightstyle
    Aliases : -hstyle
    Type : string
    Allow list : false
    Default : Color
    Simple : true
    Required : false
    Legal values : Color Stick BallAndStick Cogwheel Lasso
    Brief : Highlighting style
```

See also

- *OEHighlightParamName.Style* constant
- *OEConfigureHighlightStyle* function
- *OEGetHighlightStyle* function

4.2.19 OEHighlightStyle

This namespace contains constants representing various styles of highlighting.

See also

- *OEAddHighlighting* function
- *OEHighlightBase* class
- *Highlighting* chapter

The *OEHighlightStyle* namespace contains the following constants:

Default

The default highlighting style is *OEHighlightStyle.Color*.

BallAndStick

See example in *Figure: Example of highlighting using the 'ball and stick' style*.

- Atoms are highlighted by depicting spheres underneath them.
- Bonds are highlighted by depicting sticks underneath them.

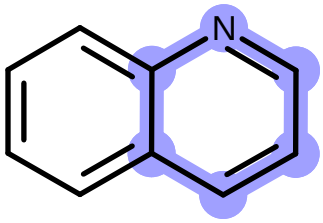


Fig. 276: **Example of highlighting using the 'ball and stick' style**

The radius of the spheres and the width of the sticks are automatically scaled with the molecule. See example in *Figure: Example of scaling the 'ball and stick' style highlighting along with the molecule*.

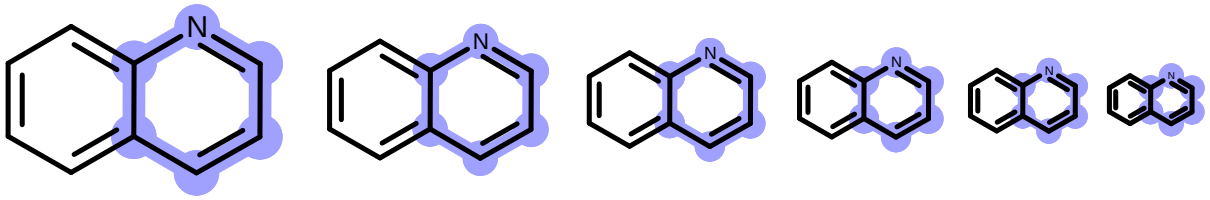


Fig. 277: **Example of scaling the 'ball and stick' style highlighting along with the molecule**

See also

- *OEHighlightByBallAndStick* class

Color

See example in *Figure: Example of highlighting using the 'color' style*.

- Atoms are highlighted by changing the color of their label, if present. If the atom being highlighted is an isolated carbon atom, then its atom label is explicitly displayed. See example in *Figure: Example of highlighting isolated atoms using the 'color' style*.
- Bonds are highlighted by changing their color.

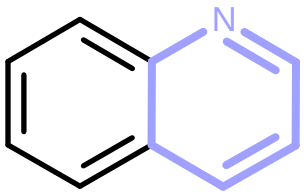


Fig. 278: **Example of highlighting using the 'color' style**

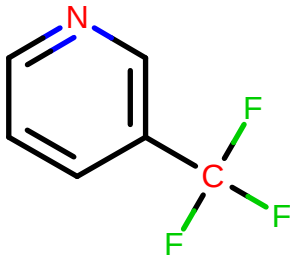


Fig. 279: **Example of highlighting isolated atoms using the 'color' style**

See also

- *OEHighlightByColor* class

Cogwheel

See example in *Figure: Example of highlighting using the 'cogwheel' style*.

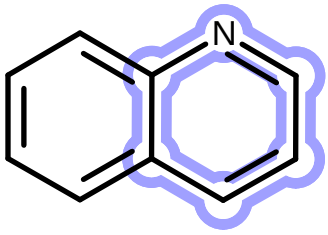


Fig. 280: **Example of highlighting using the 'cogwheel' style**

The radius of the spheres and the width of the sticks are automatically scaled with the molecule. See example in *Figure: Example of scaling the 'cogwheel' style highlighting along with the molecule*.

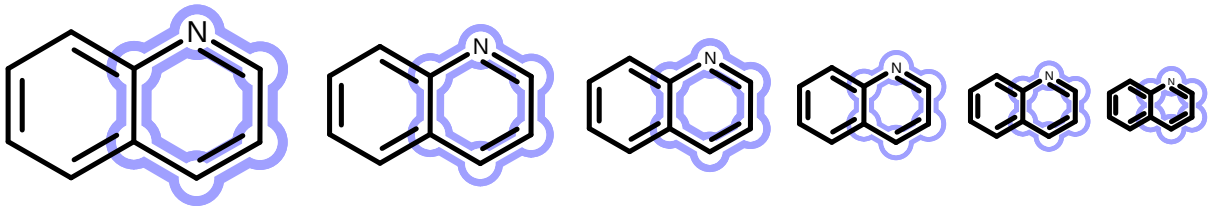


Fig. 281: **Example of scaling the 'cogwheel' style highlighting along with the molecule**

See also

- *OEHighlightByCogwheel* class

Lasso

See example in *Figure: Example of highlighting using the 'lasso' style*.

Atoms and bond are highlighted by drawing a "curved" convex hull around them.

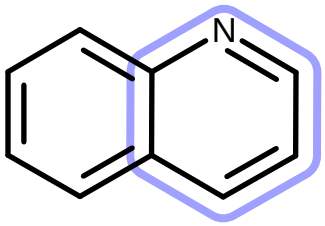


Fig. 282: **Example of highlighting using the 'lasso' style**

The width of "lasso" line is automatically scaled with the molecule. See example in *Figure: Example of scaling the 'lasso' style highlighting along with the molecule*.

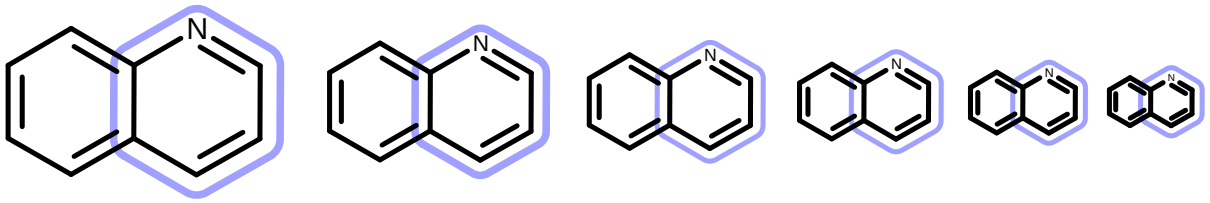


Fig. 283: **Example of scaling the 'lasso' style highlighting along with the molecule**

See also

- *OEHighlightByLasso* class

Stick

See example in *Figure: Example of highlighting using the 'stick' style*.

- Atoms are highlighted by changing the color of their label to be monochrome, if present. Isolated atoms are highlighted by depicting spheres underneath them. See example in *Figure: Example of highlighting isolated atoms using the 'stick' style*.
- Bonds are highlighted by depicting sticks underneath them.

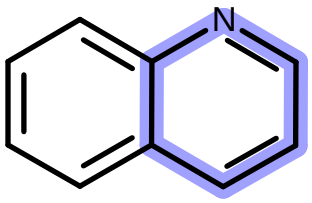


Fig. 284: **Example of highlighting using the 'stick' style**

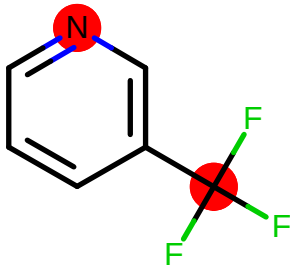


Fig. 285: **Example of highlighting isolated atoms using the 'stick' style**

The width of the sticks is automatically scaled with the molecule. See example in *Figure: Example of scaling the 'stick' style highlighting along with the molecule*.

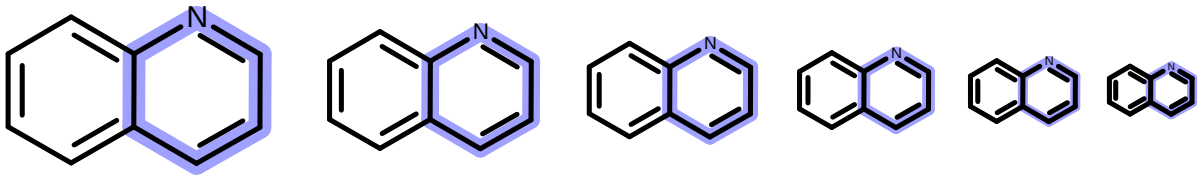


Fig. 286: **Example of scaling the 'stick' style highlighting along with the molecule**

See also

- *OEHighlightByStick* class

4.2.20 OEHydrogenPos

This namespace contains constants representing the position of hydrogen relative to the atomic label.

See also

- `OE2DAtomDisplay.GetHPosition` method
- `OE2DAtomDisplay.SetHPosition` method

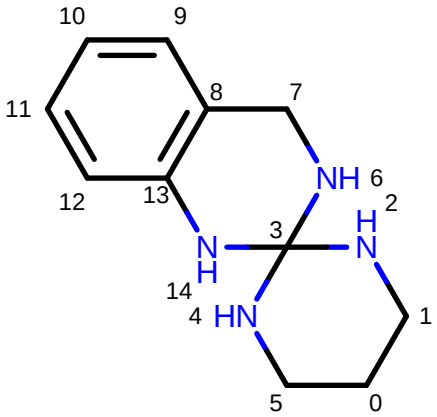


Fig. 287: Example of various hydrogen positions

The `OEHydrogenPos` namespace contains the following constants:

Undefined

The position of the hydrogen is not calculated yet.

East

See atom with index 4 in *Figure: Example of various hydrogen positions*

North

See atom with index 14 in *Figure: Example of various hydrogen positions*

South

See atom with index 2 in *Figure: Example of various hydrogen positions*

West

See atom with index 6 in *Figure: Example of various hydrogen positions*

4.2.21 OEHydrogenStyle

This namespace contains constants that control how explicit and implicit hydrogen information is depicted.

See also

- *OE2DMolDisplayOptions.GetHydrogenStyle* method
- *OE2DMolDisplayOptions.SetHydrogenStyle* method

The *OEHydrogenStyle* namespace contains the following constants:

Default

The default hydrogen display style is the combination of *OEHydrogenStyle.ImplicitHetero* and *OEHydrogenStyle.ExplicitAll*. See example in *Figure: Example of the default hydrogens depiction*.

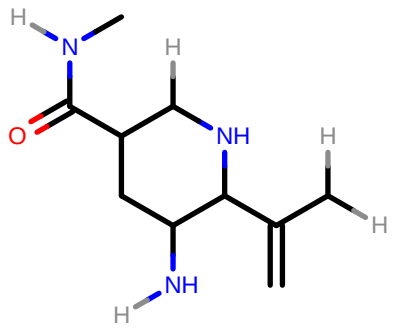


Fig. 288: **Example of the default hydrogens depiction**

ExplicitAll

Depicts all explicit hydrogens. See example in *Figure: Example of depicting all explicit hydrogens*.

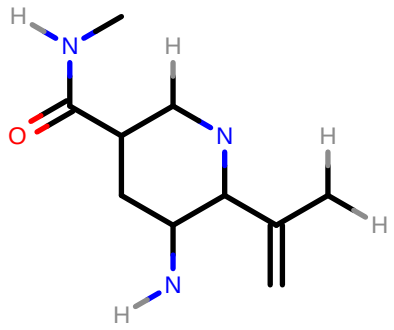


Fig. 289: **Example of depicting all explicit hydrogens**

ExplicitHetero

Depicts explicit hydrogens attached to hetero atoms. See example in *Figure: Example of depicting explicit hydrogens attached to hetero atoms*.

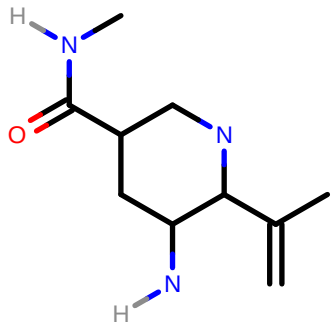


Fig. 290: **Example of depicting explicit hydrogens attached to hetero atoms**

ExplicitTerminal

Depicts explicit hydrogens attached to terminal atoms. See example in *Figure: Example of depicting explicit hydrogens attached to terminal atoms*.

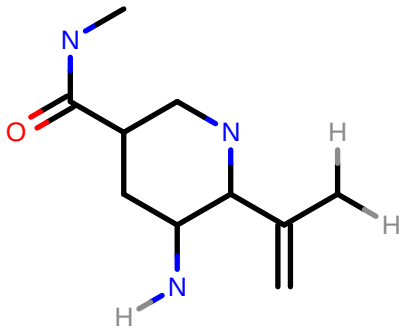


Fig. 291: **Example of depicting explicit hydrogens attached to terminal atoms**

Hidden

No hydrogen information is displayed. See example in *Figure: Example of hiding all hydrogens*.

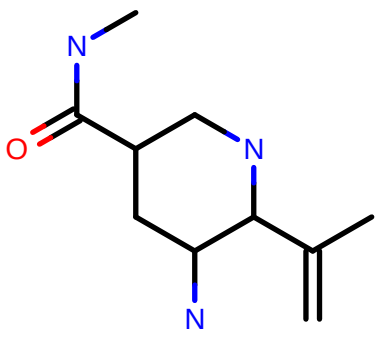


Fig. 292: **Example of hiding all hydrogens**

ImplicitAll

Depicts all implicit hydrogens. See example in *Figure: Example of depicting all implicit hydrogens*.

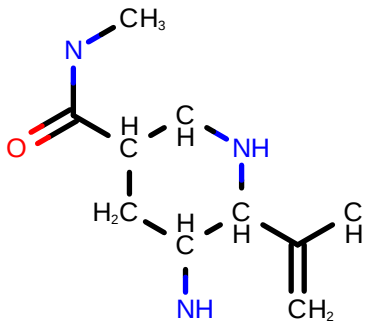


Fig. 293: **Example of depicting all implicit hydrogens**

ImplicitHetero

Depicts implicit hydrogens attached to hetero atoms. See example in *Figure: Example of depicting implicit hydrogens attached to hetero atoms*.

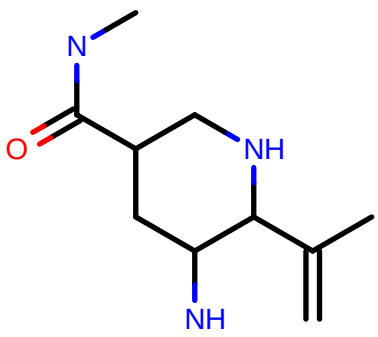


Fig. 294: **Example of depicting implicit hydrogens attached to hetero atoms**

ImplicitTerminal

Depicts implicit hydrogens attached to terminal atoms. See example in *Figure: Example of depicting implicit hydrogens attached to terminal atoms*.

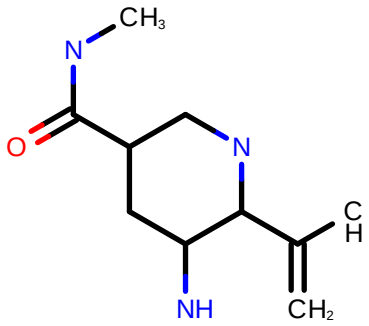


Fig. 295: **Example of depicting implicit hydrogens attached to terminal atoms**

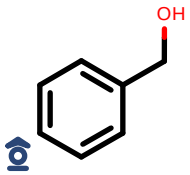
4.2.22 OEIconLocation

This namespace contains constants representing the position of the logo when using the *OEAddOpenEyeIcon* and *OEAddInteractiveIcon* functions. The *OEIconLocation* namespace contains the following constants:

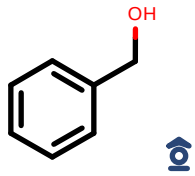
Default

The default title location is *OEIconLocation.BottomRight*.

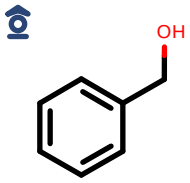
BottomLeft



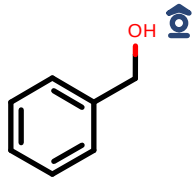
BottomRight



TopLeft



TopRight



4.2.23 OEImageGridParamName

This namespace contains the default parameter names used when the interface generated by invoking the *OEConfigureImageGridParams* function.

See also

- *OEImageGridSetup* namespace.

The *OEImageGridParamName* namespace contains the following constants:

NumColumns

The default name of number of grid columns parameter is **"-cols"**.

See also

- *OEImageGridSetup.NumColumns* constant
- *OEConfigureImageGridNumColumns* function
- *OEGetImageGridNumColumns* function

NumRows

The default name of number of grid rows parameter is **"-rows"**.

See also

- *OEImageGridSetup.NumRows* constant
- *OEConfigureImageGridNumRows* function
- *OEGetImageGridNumRows* function

4.2.24 OEImageGridSetup

This namespace encodes symbolic constants used as bit-masks to indicate which image grid interface parameters are being created when invoking the *OEConfigureImageGridParams* function.

See also

- *OEImageGridParamName* namespace.

The *OEImageGridSetup* namespace contains the following constants:

All

The combination of following constants:

- *OEImageGridSetup.NumColumns*
- *OEImageGridSetup.NumRows*

NumColumns

Passing this constant to the *OEConfigureImageGridParams* function result in generating the following default interface to configure the number of columns of image grid by invoking the *OEConfigureImageGridNumColumns* function.

```
Contents of parameter -cols
    Aliases : -cols
    Type : int
    Allow list : false
    Default : 2
    Simple : true
    Required : false
    Legal ranges :
        10 to 1
    Brief : Number of columns
```

See also

- *OEImageGridParamName.NumColumns* constant
- *OEConfigureImageGridNumColumns* function
- *OEGetImageGridNumColumns* function

NumRows

Passing this constant to the *OEConfigureImageGridParams* function result in generating the following default interface to configure the number of row of image grid by invoking the *OEConfigureImageGridNumRows* function.

```
Contents of parameter -rows
    Aliases : -rows
    Type : int
    Allow list : false
    Default : 3
    Simple : true
    Required : false
    Legal ranges :
        10 to 1
    Brief : Number of rows
```

See also

- *OEImageGridParamName.NumRows* constant
- *OEConfigureImageGridNumRows* function
- *OEGetImageGridNumRows* function

4.2.25 OEImageParamName

This namespace contains the default parameter names used when the interface generated by invoking the *OEConfigureImageOptions* function.

See also

- *OEImageSetup* namespace.

The *OEMultiPageParamName* namespace contains the following constants:

Height

The default name of the image height parameter is **"-height"**.

See also

- *OEImageSetup.Height* constant
- *OEConfigureImageOptions* function
- *OEGetImageHeight* function

Width

The default name of the image width parameter is **"-width"**.

See also

- *OEImageSetup.Width* constant
- *OEConfigureImageOptions* function
- *OEGetImageWidth* function

4.2.26 OEImageSetup

This namespace encodes symbolic constants used as bit-masks to indicate which image interface parameters are being created when invoking the *OEConfigureImageOptions* function.

See also

- *OEImageParamName* namespace.

The *OEImageSetup* namespace contains the following constants:

All

The combination of following constants:

- *OEImageSetup.Width*
- *OEImageSetup.Height*

Height

Passing this constant to the *OEConfigureImageOptions* function result in generating the following default interface to configure image width.

```
Contents of parameter -width
    Aliases : -w
    Type : int
    Allow list : false
    Default : 0
    Simple : true
    Required : false
    Legal ranges :
        5000 to 0
    Brief : Width of output image
```

See also

- *OEImageParamName.Height* function
- *OEConfigureImageOptions* function
- *OEGetImageHeight* function

Width

Passing this constant to the *OEConfigureImageOptions* function result in generating the following default interface to configure image width.

```
Contents of parameter -height
    Aliases : -h
    Type : int
    Allow list : false
    Default : 0
    Simple : true
    Required : false
    Legal ranges :
```

(continues on next page)

(continued from previous page)

```
      5000 to 0
  Brief : Height of output image
```

See also

- *OEImageParamName.Width* function
- *OEConfigureImageOptions* function
- *OEGetImageWidth* function

4.2.27 OEImageTableStyle

This namespace contains constants that define built-in image table styles.

See also

- *OEImageTable* class
- *OEImageTableOptions* class

Default

The default image table style is *OEImageTableStyle.NoStyle*.

LightBlue

(1, 1)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)
(4, 1)	(4, 2)	(4, 3)
(5, 1)	(5, 2)	(5, 3)

row 1	(1, 1)	(1, 2)
row 2	(2, 1)	(2, 2)
row 3	(3, 1)	(3, 2)
row 4	(4, 1)	(4, 2)
row 5	(5, 1)	(5, 2)

column 1	column 2	column 3
(1, 1)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)
(4, 1)	(4, 2)	(4, 3)

	column 1	column 2
row 1	(1, 1)	(1, 2)
row 2	(2, 1)	(2, 2)
row 3	(3, 1)	(3, 2)
row 4	(4, 1)	(4, 2)

Fig. 296: **Examples of using the 'LightBlue' image table style**

LightGreen

(1, 1)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)
(4, 1)	(4, 2)	(4, 3)
(5, 1)	(5, 2)	(5, 3)

row 1	(1, 1)	(1, 2)
row 2	(2, 1)	(2, 2)
row 3	(3, 1)	(3, 2)
row 4	(4, 1)	(4, 2)
row 5	(5, 1)	(5, 2)

column 1	column 2	column 3
(1, 1)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)
(4, 1)	(4, 2)	(4, 3)

	column 1	column 2
row 1	(1, 1)	(1, 2)
row 2	(2, 1)	(2, 2)
row 3	(3, 1)	(3, 2)
row 4	(4, 1)	(4, 2)

Fig. 297: **Examples of using the 'LightGreen' image table style**

LightGrey

(1, 1)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)
(4, 1)	(4, 2)	(4, 3)
(5, 1)	(5, 2)	(5, 3)

row 1	(1, 1)	(1, 2)
row 2	(2, 1)	(2, 2)
row 3	(3, 1)	(3, 2)
row 4	(4, 1)	(4, 2)
row 5	(5, 1)	(5, 2)

column 1	column 2	column 3
(1, 1)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)
(4, 1)	(4, 2)	(4, 3)

	column 1	column 2
row 1	(1, 1)	(1, 2)
row 2	(2, 1)	(2, 2)
row 3	(3, 1)	(3, 2)
row 4	(4, 1)	(4, 2)

Fig. 298: **Examples of using the 'LightGrey' image table style**

MediumBlue

(1, 1)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)
(4, 1)	(4, 2)	(4, 3)
(5, 1)	(5, 2)	(5, 3)

row 1	(1, 1)	(1, 2)
row 2	(2, 1)	(2, 2)
row 3	(3, 1)	(3, 2)
row 4	(4, 1)	(4, 2)
row 5	(5, 1)	(5, 2)

column 1	column 2	column 3
(1, 1)	(1, 2)	(1, 3)
(2, 1)	(2, 2)	(2, 3)
(3, 1)	(3, 2)	(3, 3)
(4, 1)	(4, 2)	(4, 3)

	column 1	column 2
row 1	(1, 1)	(1, 2)
row 2	(2, 1)	(2, 2)
row 3	(3, 1)	(3, 2)
row 4	(4, 1)	(4, 2)

Fig. 299: **Examples of using the 'MediumBlue' image table style**

MediumGreen

(1, 1)	(1, 2)	(1, 3)	row 1	(1, 1)	(1, 2)	column 1	column 2	column 3		column 1	column 2
(2, 1)	(2, 2)	(2, 3)	row 2	(2, 1)	(2, 2)	(1, 1)	(1, 2)	(1, 3)	row 1	(1, 1)	(1, 2)
(3, 1)	(3, 2)	(3, 3)	row 3	(3, 1)	(3, 2)	(2, 1)	(2, 2)	(2, 3)	row 2	(2, 1)	(2, 2)
(4, 1)	(4, 2)	(4, 3)	row 4	(4, 1)	(4, 2)	(3, 1)	(3, 2)	(3, 3)	row 3	(3, 1)	(3, 2)
(5, 1)	(5, 2)	(5, 3)	row 5	(5, 1)	(5, 2)	(4, 1)	(4, 2)	(4, 3)	row 4	(4, 1)	(4, 2)

Fig. 300: **Examples of using the 'MediumGreen' image table style**

MediumGrey

(1, 1)	(1, 2)	(1, 3)	row 1	(1, 1)	(1, 2)	column 1	column 2	column 3		column 1	column 2
(2, 1)	(2, 2)	(2, 3)	row 2	(2, 1)	(2, 2)	(1, 1)	(1, 2)	(1, 3)	row 1	(1, 1)	(1, 2)
(3, 1)	(3, 2)	(3, 3)	row 3	(3, 1)	(3, 2)	(2, 1)	(2, 2)	(2, 3)	row 2	(2, 1)	(2, 2)
(4, 1)	(4, 2)	(4, 3)	row 4	(4, 1)	(4, 2)	(3, 1)	(3, 2)	(3, 3)	row 3	(3, 1)	(3, 2)
(5, 1)	(5, 2)	(5, 3)	row 5	(5, 1)	(5, 2)	(4, 1)	(4, 2)	(4, 3)	row 4	(4, 1)	(4, 2)

Fig. 301: **Examples of using the 'MediumGrey' image table style**

NoStyle

(1, 1)	(1, 2)	(1, 3)	row 1	(1, 1)	(1, 2)	column 1	column 2	column 3		column 1	column 2
(2, 1)	(2, 2)	(2, 3)	row 2	(2, 1)	(2, 2)	(1, 1)	(1, 2)	(1, 3)	row 1	(1, 1)	(1, 2)
(3, 1)	(3, 2)	(3, 3)	row 3	(3, 1)	(3, 2)	(2, 1)	(2, 2)	(2, 3)	row 2	(2, 1)	(2, 2)
(4, 1)	(4, 2)	(4, 3)	row 4	(4, 1)	(4, 2)	(3, 1)	(3, 2)	(3, 3)	row 3	(3, 1)	(3, 2)
(5, 1)	(5, 2)	(5, 3)	row 5	(5, 1)	(5, 2)	(4, 1)	(4, 2)	(4, 3)	row 4	(4, 1)	(4, 2)

Fig. 302: **Examples of using the 'NoStyle' image table style**

SantaFe

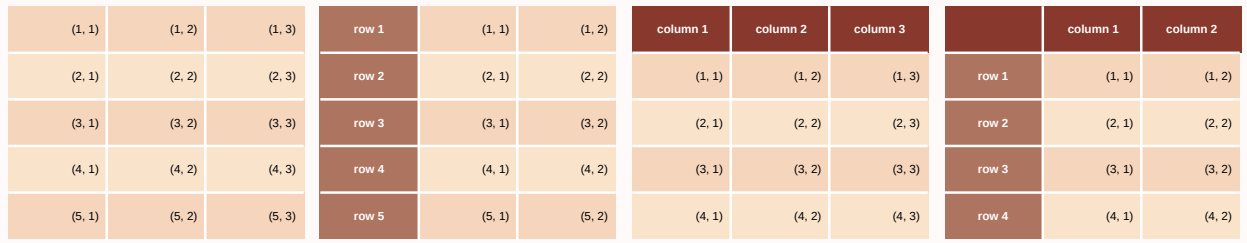


Fig. 303: **Examples of using the 'SantaFe' image table style**

4.2.28 OEInteractiveEffect

This namespace contains constants related to interactive effects when generating SVG images.

None

No interactive effect.

Hover

Interactive effect triggered when hovering over some area in the image.

Toggle

Interactive effect triggered when clicking on some area in the image.

4.2.29 OELayerPosition

This namespace contains constants representing the position of layers available for an *OE2DMolDisplay* class.

See also

- *OE2DMolDisplay.GetLayer*

The *OELayerPosition* namespace contains the following constants:

Above

Provides access to **above** image (*OEImage*) of a *OE2DMolDisplay* object. Everything drawn to this image will be rendered last when calling the *OERenderMolecule* function, as a result the image will appear **above** the molecule depiction. See example in *Figure: Example of drawing a triangle into the above layer*.

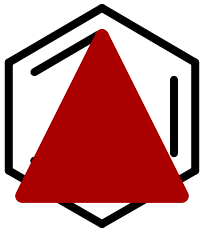


Fig. 304: **Example of drawing a triangle into the above layer**

Below

Provides access to **below** image (*OEImage*) of a *OE2DMolDisplay* object. Everything drawn to this image will be rendered first when calling the `OERenderMolecule` function, as a result the image will appear **below** the molecule depiction.

For example, the sticks drawn to highlighting bonds when using the `OEHighlightStyle.Stick` style all drawn into a *below* layer. See example in *Figure: Example of drawing a triangle into the below layer*.

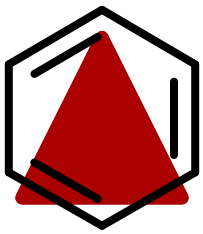


Fig. 305: **Example of drawing a triangle into the below layer**

SVG

This layer is used internally when generating interactive SVG images.

4.2.30 OELegendColorStyle

This namespace contains constants that define built-in legend color styles.

See also

- *OELegendLayout* class
- *OELegendLayoutOptions* class
- `OELegendLayoutStyle` namespace
- `OELegendInteractiveStyle` namespace

Default

The default legend color style is *OELegendColorStyle.NoStyle*.

LightBlue



Fig. 306: **Example of using the 'LightBlue' color style**

LightGreen



Fig. 307: **Example of using the 'LightGreen' color style**

LightGrey



Fig. 308: **Example of using the 'LightGrey' color style**

NoStyle



Fig. 309: **Example of using the 'NoStyle' color style**

4.2.31 OELegendInteractiveStyle

This namespace contains constants that define built-in legend interactive styles.

See also

- *OELegendLayout* class
- *OELegendLayoutOptions* class
- *OELegendLayoutStyle* namespace
- *OELegendColorStyle* namespace

Default

The default legend interactive style is *OELegendInteractiveStyle.Hover*.

Hover

> **Note**
>
> This style is **only available** for `.svg` image format.

Permanent

Toggle

> **Note**
>
> This style is **only available** for `.svg` image format.

4.2.32 OELegendLayoutStyle

This namespace contains constants that define built-in legend layout styles.

> **See also**
>
> - *OELegendLayout* class
> - *OELegendLayoutOptions* class
> - `OELegendColorStyle` namespace
> - `OELegendInteractiveStyle` namespace

Default

The default legend layout style is `OELegendLayoutStyle.HorizontalTopLeft`.

HorizontalBottomLeft



Fig. 310: **Example of using the 'HorizontalBottomLeft' layout style**

HorizontalBottomRight



Fig. 311: **Example of using the 'HorizontalBottomRight' layout style**

HorizontalTopLeft

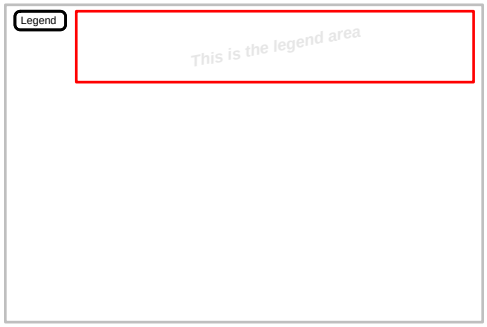


Fig. 312: **Example of using the 'HorizontalTopLeft' layout style**

HorizontalTopRight



Fig. 313: **Example of using the 'HorizontalTopRight' layout style**

VerticalBottomLeft



Fig. 314: **Example of using the 'VerticalBottomLeft' layout style**

VerticalBottomRight



Fig. 315: **Example of using the 'VerticalBottomRight' layout style**

VerticalTopLeft



Fig. 316: **Example of using the 'VerticalTopLeft' layout style**

VerticalTopRight



Fig. 317: **Example of using the 'VerticalTopRight' layout style**

4.2.33 OELineCap

This namespace contains constants that control how the endpoints of lines are drawn.

See also

- *OEPen.GetLineCap* method
- *OEPen.SetLineCap* method

The *OELineCap* namespace contains the following constants:

Default

The default line cap style is *OELineCap.Round*.

Butt

Closes the line off with a straight edge that's normal (at 90 degrees to) the direction of the stroke and crosses its end. See example in *Figure: Example of line drawing with 'Butt' line cap style*.



Fig. 318: **Example of line drawing with 'Butt' line cap style**

Round

Produces a rounded effect on the end of the line. See example in *Figure: Example of line drawing with 'Round' line cap style*.



Fig. 319: **Example of line drawing with 'Round' line cap style**

Square

Has the same appearance as the `OELineCap.Butt` style, but stretches the line slightly beyond the actual path. See example in *Figure: Example of line drawing with 'Square' line cap style*.



Fig. 320: **Example of line drawing with 'Square' line cap style**

4.2.34 OELineJoin

The line join style controls how lines connect at corners when drawing outlines.

See also

- `OEPen.GetLineJoin` method
- `OEPen.SetLineJoin` method

The `OELineJoin` namespace contains the following constants:

Default

The default line join style is *OELineJoin.Round*.

Bevel

See example in *Figure: Example of line drawing with 'Bevel' line join style*.



Fig. 321: **Example of line drawing with 'Bevel' line join style**

Miter

See example in *Figure: Example of line drawing with 'Miter' line join style*.

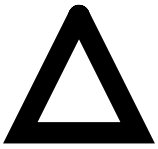


Fig. 322: **Example of line drawing with 'Miter' line join style**

Round

See example in *Figure: Example of line drawing with 'Round' line join style*.



Fig. 323: **Example of line drawing with 'Round' line join style**

4.2.35 OEMargin

This namespace contains constants representing the position of margins for the *OEImageGrid* and the *OE2DMolDisplayOptions* classes .

See also

- *OE2DMolDisplayOptions.SetMargin*
- *OE2DMolDisplayOptions.GetMargin*
- *OEImageGrid.SetMargin*
- *OEImageGrid.GetMargin*

The *OEMargin* namespace contains the following constants:

Top

Bottom

Left

Right

4.2.36 OEMultiPageParamName

This namespace contains the default parameter names used when the interface generated by invoking the *OEConfigureMultiPageParams* function.

See also

- *OEMultiPageSetup* namespace.

The *OEMultiPageParamName* namespace contains the following constants:

PageOrientation

The default name of the page orientation parameter is **"-pageorientation"**.

See also

- *OEMultiPageSetup.PageOrientation* constant
- *OEConfigureMultiPageOrientation* function
- *OEGetMultiPageOrientation* function

PageSize

The default name of the page size parameter is **"-pagesize"**.

See also

- *OEMultiPageSetup.PageSize* constant
- *OEConfigureMultiPageSize* function
- *OEGetMultiPageSize* function

4.2.37 OEMultiPageSetup

This namespace encodes symbolic constants used as bit-masks to indicate which multi page interface parameters are being created when invoking the *OEConfigureMultiPageParams* function.

See also

- *OEMultiPageParamName* namespace.

The *OEMultiPageSetup* namespace contains the following constants:

All

The combination of following constants:

- *OEMultiPageSetup.PageOrientation*
- *OEMultiPageSetup.PageSize*

PageOrientation

Passing this constant to the *OEConfigureMultiPageParams* function result in generating the following default interface to configure the page orientation by invoking the *OEConfigureMultiPageOrientation* function.

```
Contents of parameter -pageorientation
    Aliases : -pori
    Type : string
    Allow list : false
    Default : Portrait
    Simple : true
    Required : false
    Legal values : Portrait Landscape
    Brief : Page orientation
```

See also

- *OEMultiPageParamName.PageOrientation* constant
- *OEConfigureMultiPageOrientation* function
- *OEGetMultiPageOrientation* function

PageSize

Passing this constant to the *OEConfigureMultiPageParams* function result in generating the following default interface to configure the page size by invoking the *OEConfigureMultiPageSize* function.

```
Contents of parameter -pagesize
    Aliases : -psize
    Type : string
    Allow list : false
    Default : US_Letter
    Simple : true
    Required : false
    Legal values : ISO_A4 US_Letter US_Legal Custom
    Brief : Page size
```

See also

- *OEMultiPageParamName.PageSize* constant
- *OEConfigureMultiPageSize* function
- *OEGetMultiPageSize* function

4.2.38 OEPageOrientation

This namespace contains constants representing two page orientations defined for multi-page images.

See also

- *OEMultiPageImageFile* class

The *OEPageSize* namespace contains the following constants:

Default

The default page orientation is *OEPageOrientation.Portrait*.

Portrait

Orientation where the height of the page is greater than its width.

Landscape

Orientation where the width of the page is greater than its height.

4.2.39 OEPageSize

This namespace contains constants representing various page sizes defined for multi-page images.

See also

- *OEMultiPageImageFile* class
- `OEGetPageWidth` function
- `OEGetPageHeight` function

The `OEPageSize` namespace contains the following constants:

Default

The default page size is `OEPageSize.US_Letter`.

ISO_A4

Dimensions: 210 mm × 297 mm (8.27 in × 11.69 in)

US_Letter

Dimensions: 216 mm × 279 mm (8.5 in × 11.0 in)

US_Legal

Dimension: 216 mm × 356 mm (8.5 in × 14.0 in)

Custom

Custom size

4.2.40 OEPrepareDepictionSetup

This namespace encodes symbolic constants used as bit-masks to indicate which parameters are being created when invoking the `OEConfigurePrepareDepictionOptions` function.

See also

- *OEPrepareDepictionOptions* class
- `OEPrepareDepiction` function

All

The combination of following constants:

- *OEPrepareDepictionSetup.ClearCoords*
- *OEPrepareDepictionSetup.DepictOrientation*
- *OEPrepareDepictionSetup.SuppressHydrogens*

ClearCoords

Passing this constant to the *OEConfigurePrepareDepictionOptions* result in generating the following default interface to configure whether the 2D coordinates of the molecule have to cleared and re-generated prior to molecule depiction.

```
Contents of parameter -clearcoords
    Aliases : -clear
    Type : bool
    Allow list : false
    Default : false
    Simple : true
    Required : false
    Brief : Clears and regenerates 2D coordinates of depicted molecule
```

DepictOrientation

Passing this constant to the *OEConfigurePrepareDepictionOptions* result in generating the following default interface to configure the preferred orientation of the 2D coordinates of the molecules.

```
Contents of parameter -orientation
    Aliases : -orient
    Type : string
    Allow list : false
    Default : Original
    Simple : true
    Required : false
    Legal values : Original Square Vertical Horizontal
    Brief : Set the preferred orientation of 2D coordinates
    Detail
        Original     - no-op
        Square       - orient the molecule for square images
        Vertical     - orient the molecule vertically
        Horizontal   - orient the molecule horizontally
```

SuppressHydrogens

Passing this constant to the *OEConfigurePrepareDepictionOptions* result in generating the following default interface to configure whether the hydrogens have to be suppressed prior to molecule depiction.

```
Contents of parameter -suppressH
    Aliases : -sh
    Type : bool
    Allow list : false
    Default : false
    Simple : true
    Required : false
    Brief :  Suppress explicit hydrogens of depicted molecule
    Detail
        Explicit hydrogens that are necessary to represent tetrahedral stereochemistry␣
↪are kept
```

4.2.41 OEProtectiveGroupStyle

This namespace contains constants representing the various protective group styles supported by **OEDepict TK**.

See also

- *OE2DMolDisplayOptions.GetProtectiveGroupStyle* method
- *OE2DMolDisplayOptions.SetProtectiveGroupStyle* method
- *OESuperAtomStyle* namespace

The *OEProtectiveGroupStyle* namespace contains the following constants. For some protective groups, more than one abbreviation can be used. The default is listed first.

Default

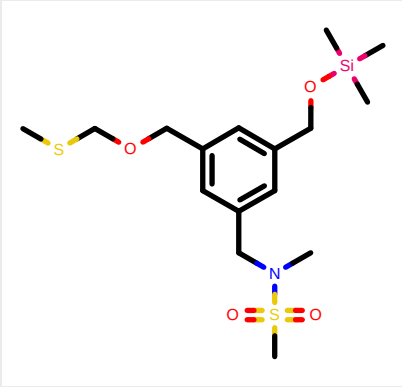
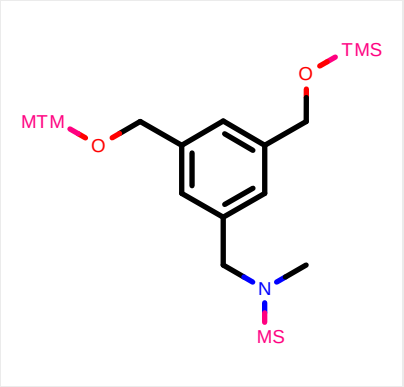
The default protective group style is *OEProtectiveGroupStyle.Off*.

Off

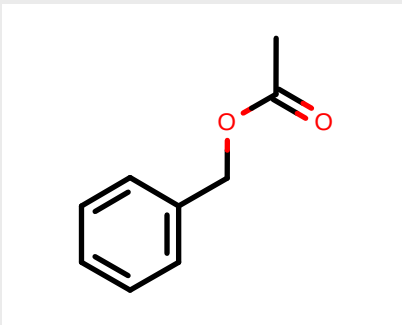
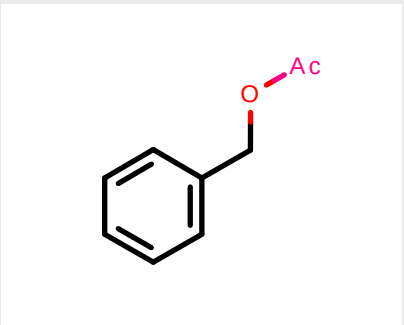
No structural abbreviation is performed.

All

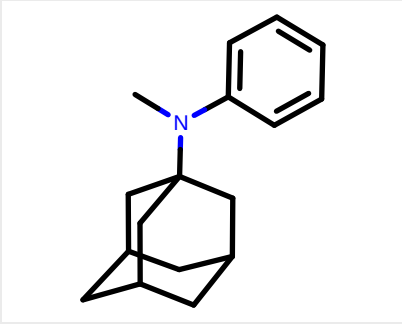
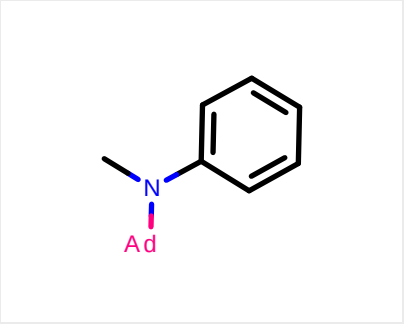
The combination of all protective group flags.

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.All*
	

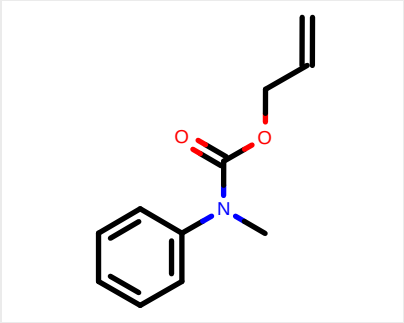
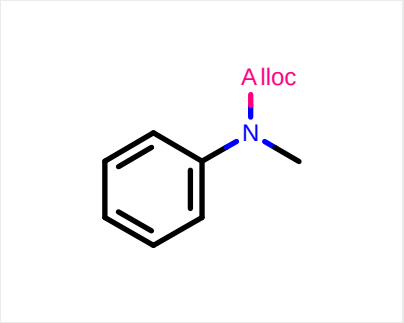
Ac

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Ac*
	

Ad

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Ad*
	

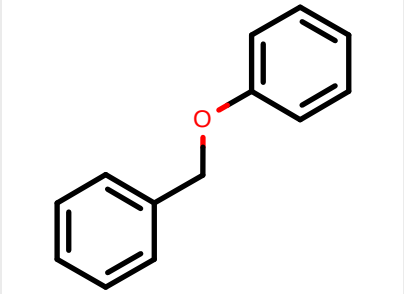
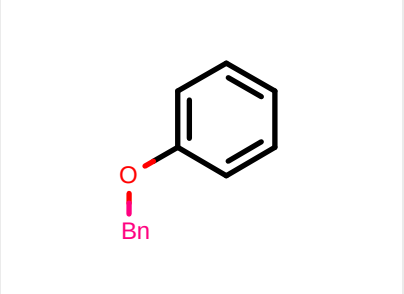
Alloc

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Alloc*
	

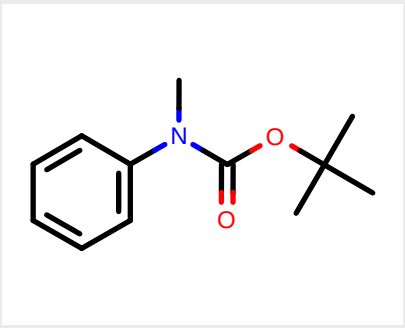
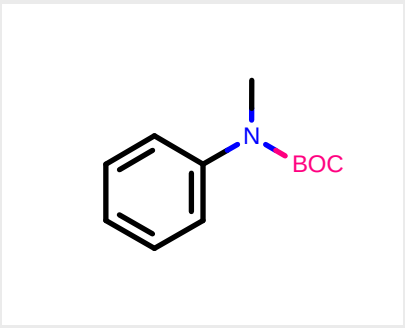
Bn

Bzl

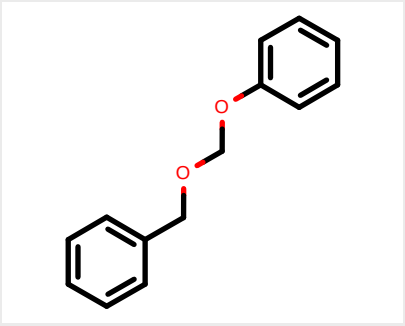
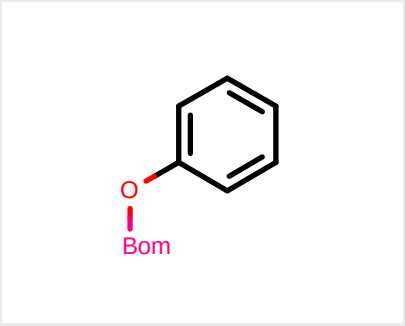
Benzyl

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Bn* *OEProtectiveGroupStyle.Bzl* *OEProtectiveGroupStyle.Benzyl*
	

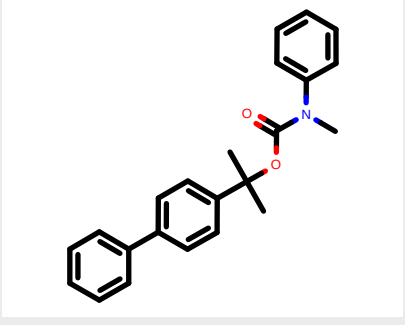
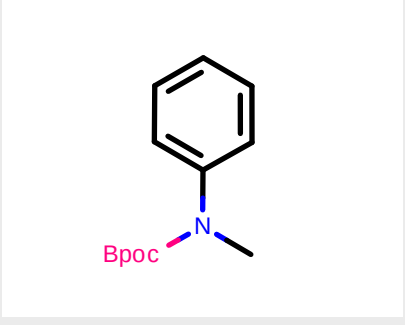
Boc

OEProtectiveGroupStyle.Off	*Boc*
	

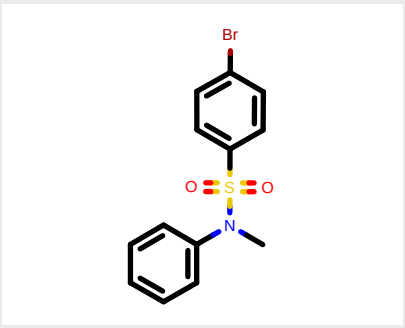
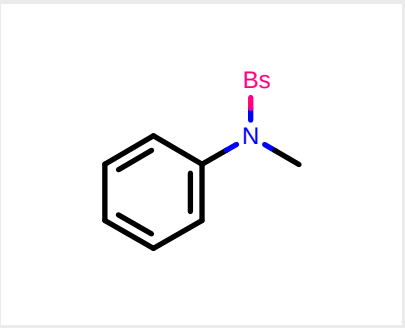
Bom

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Bom*
	

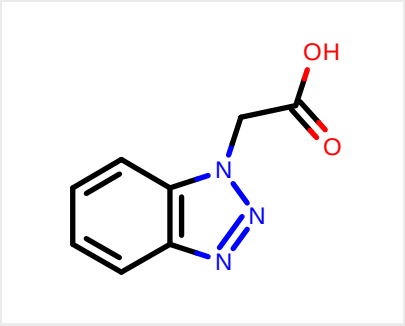
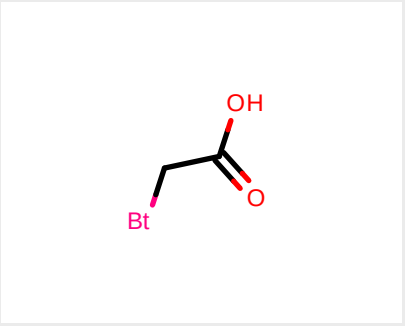
Bpoc

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Bpoc*
	

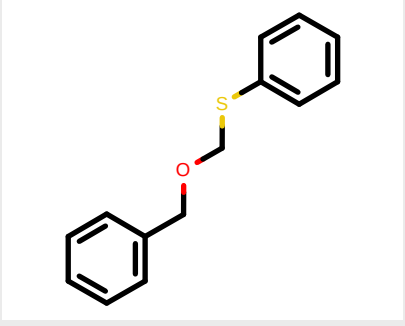
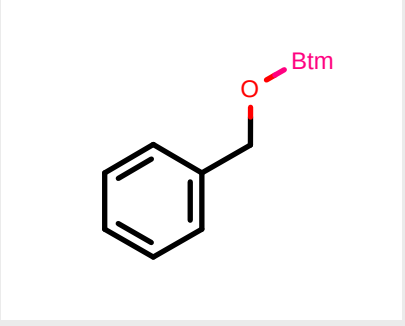
Bs

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Bs*
	

Bt

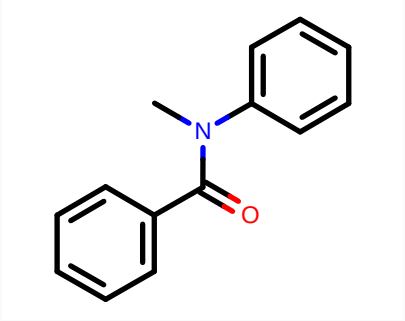
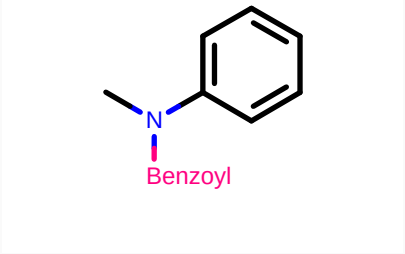
OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Bt*
	

Btm

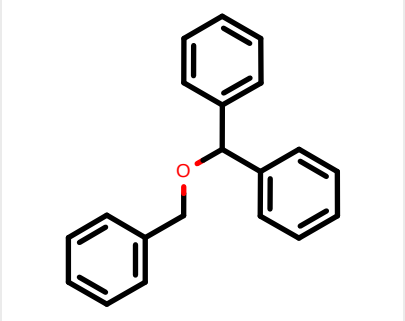
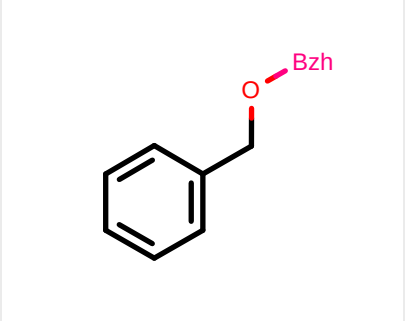
OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Btm*
	

Bz

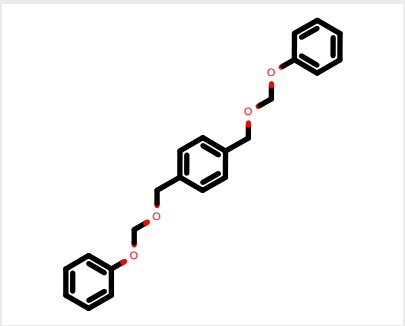
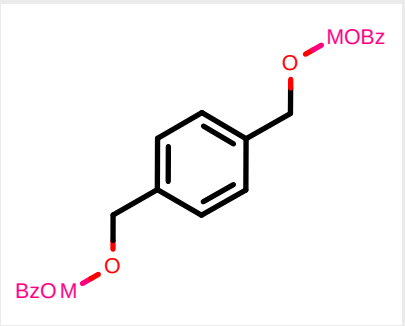
Benzoyl

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Bz*
	OEProtectiveGroupStyle.Benzoyl
	

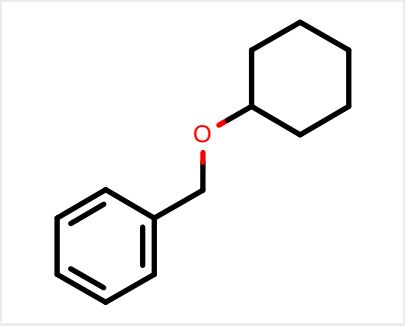
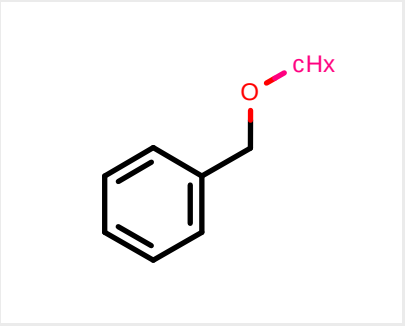
Bzh

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Bzh*
	

BzOM

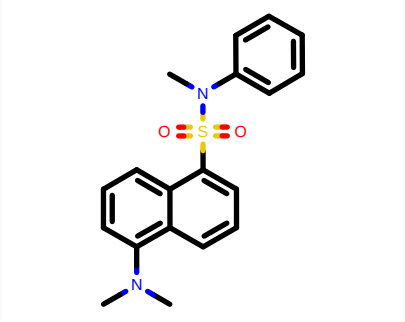
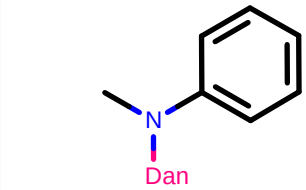
OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.BzOM*
	

cHx

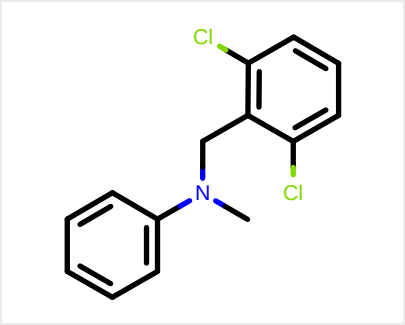
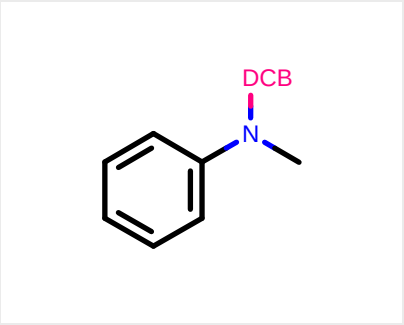
OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.cHx*
	

Dan

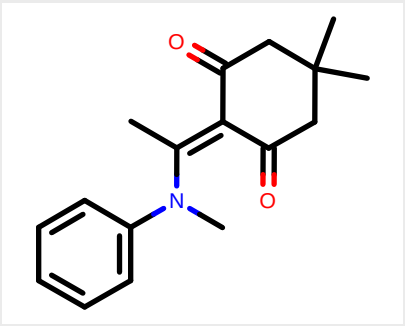
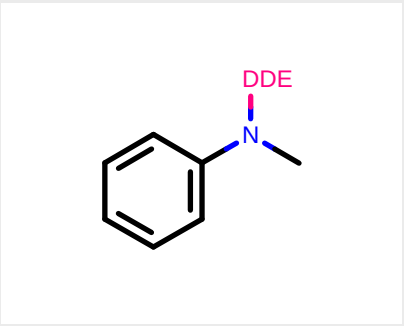
Dansyl

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Dan*
	OEProtectiveGroupStyle.Dansyl
	

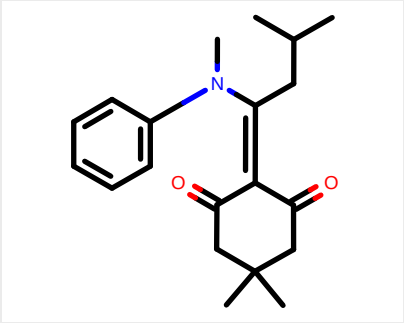
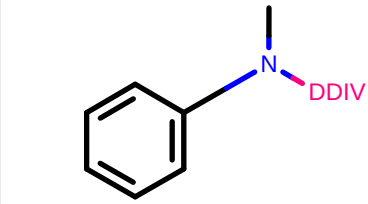
DCB

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DCB*
	

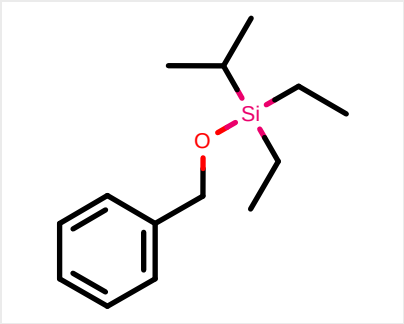
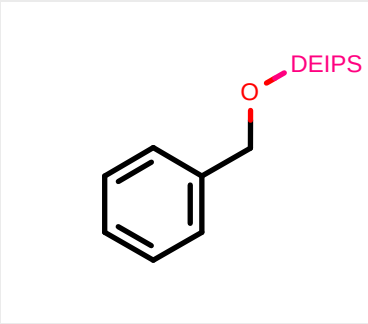
DDE

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DDE*
	

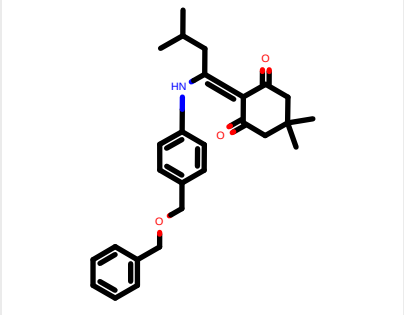
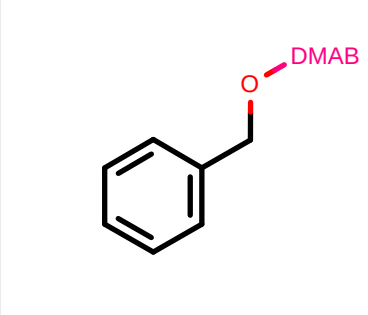
DDIV

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DDIV*
	

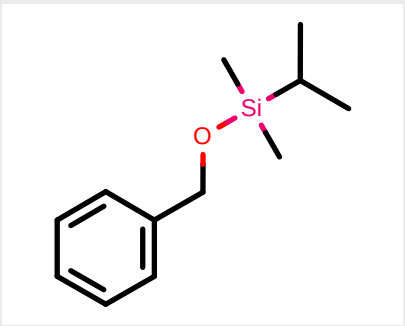
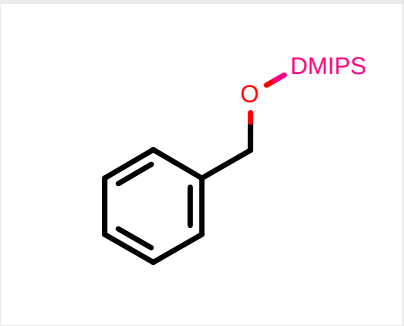
DEIPS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DEIPS*
	

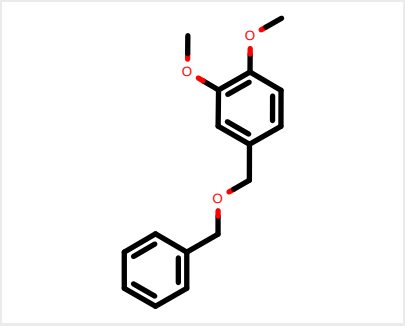
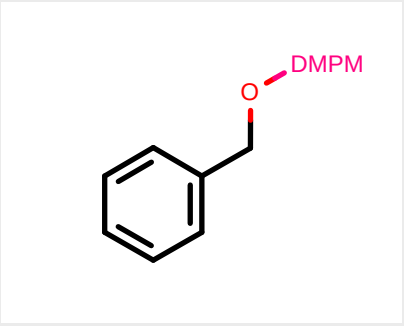
DMAB

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DMAB*
	

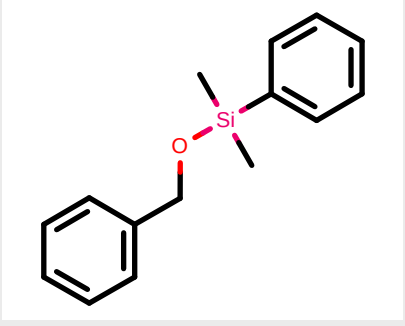
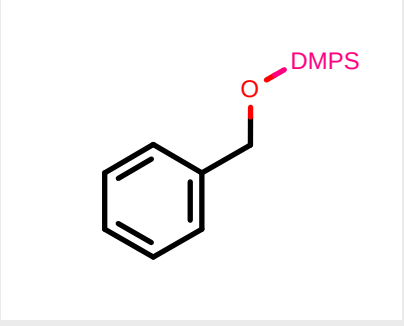
DMIPS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DMIPS*
	

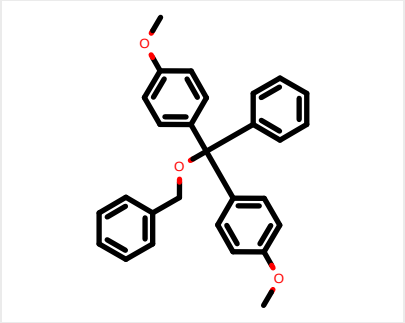
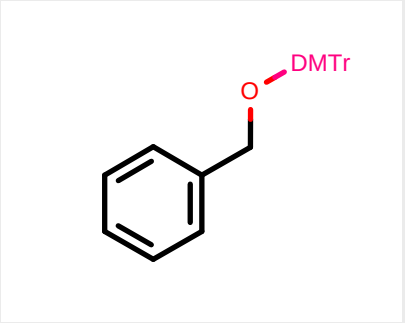
DMPM

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DMPM*
	

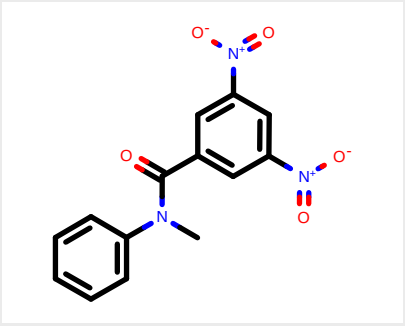
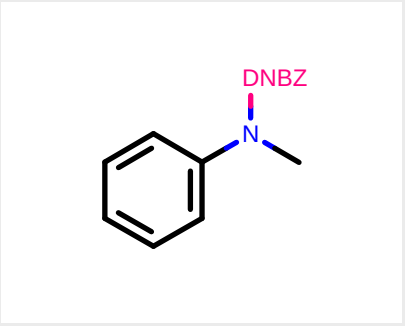
DMPS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DMPS*
	

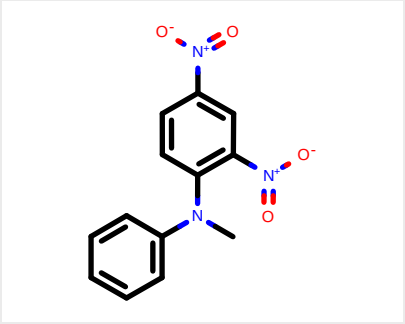
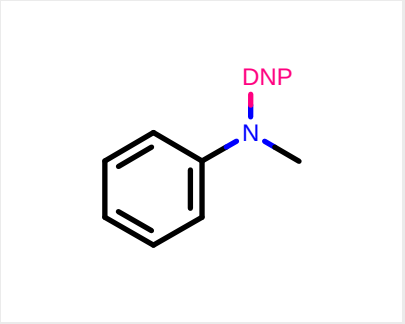
DMTr

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DMTr*
	

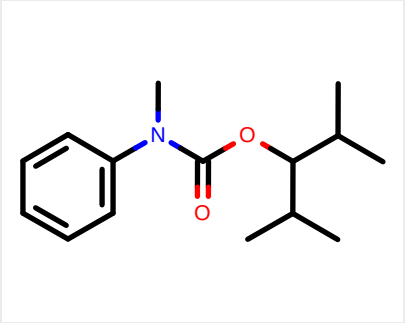
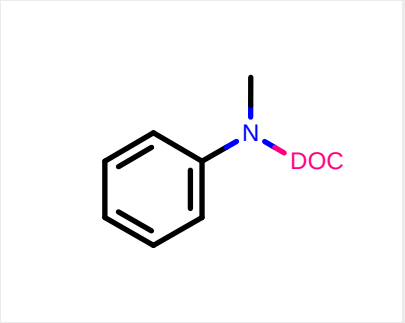
DNBZ

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DNBZ*
	

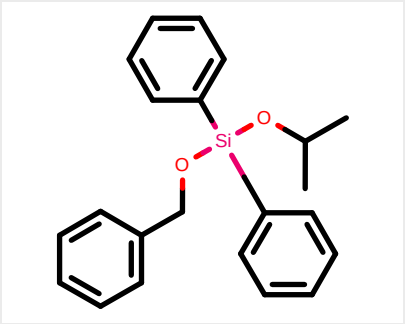
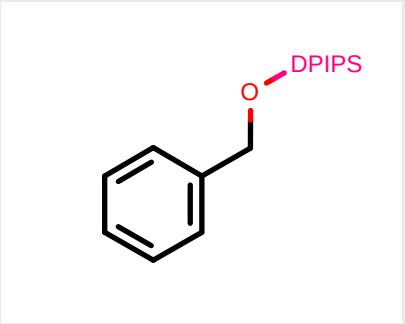
DNP

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DNP*
	

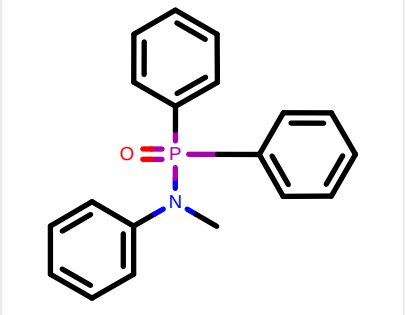
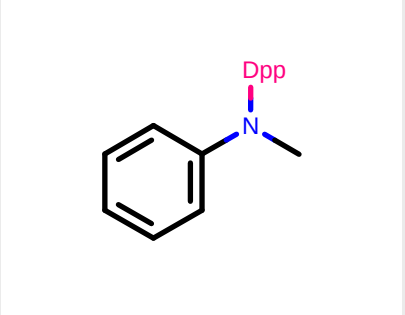
DOC

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DOC*
	

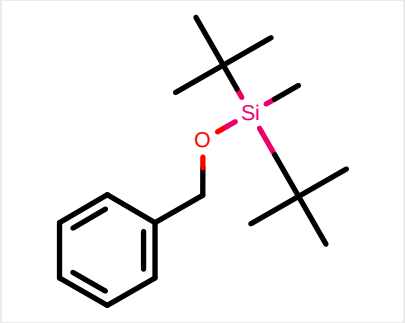
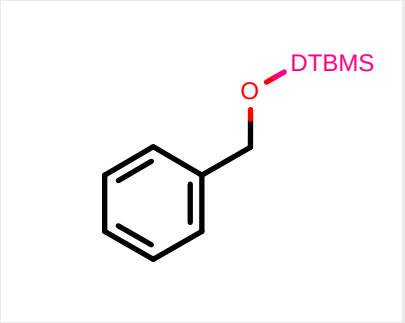
DPIPS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DPIPS*
	

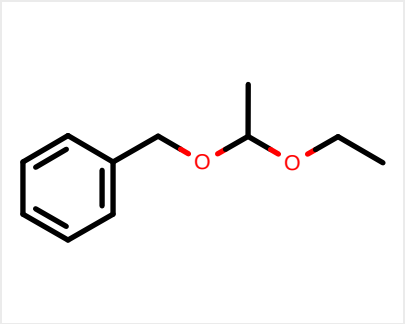
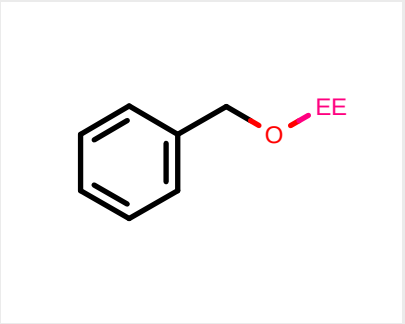
Dpp

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Dpp*
	

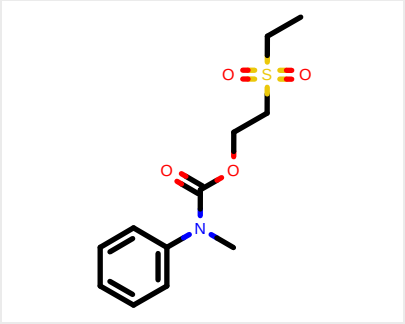
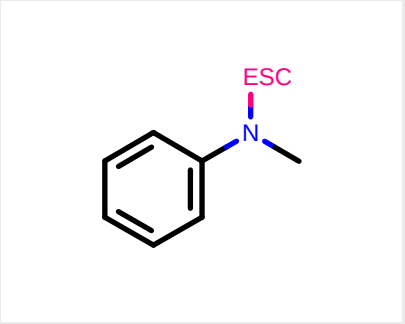
DTBMS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.DTBMS*
	

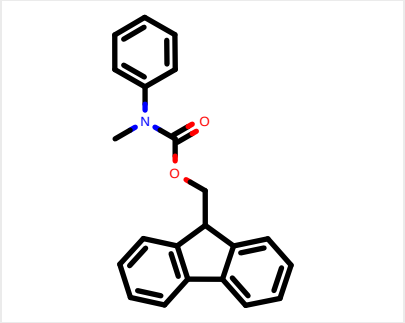
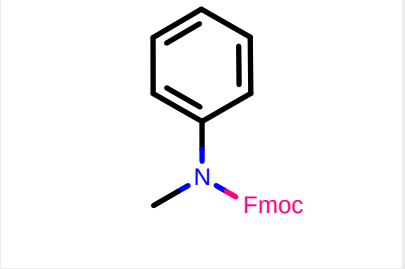
EE

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.EE*
	

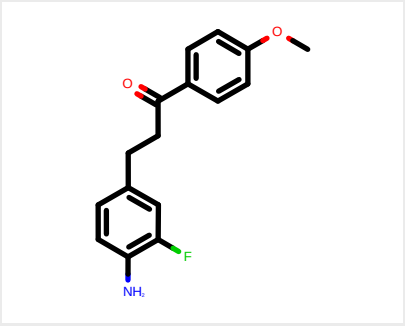
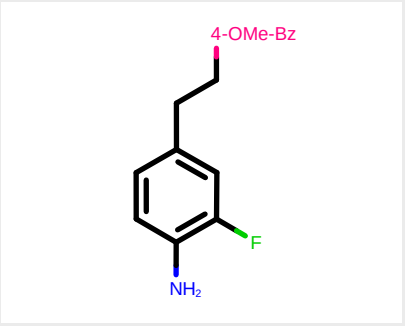
ESC

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.ESC*
	

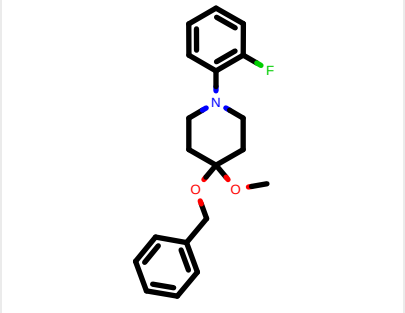
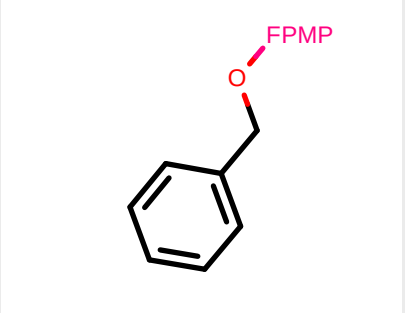
Fmoc

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Fmoc*
	

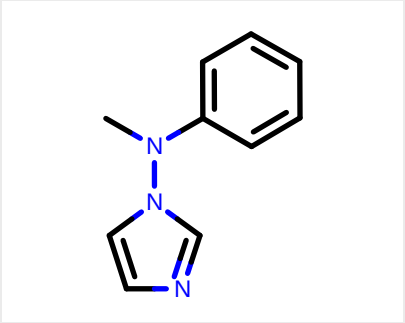
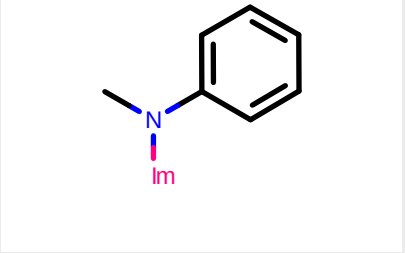
FourOMeBz

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.FourOMeBz*
	

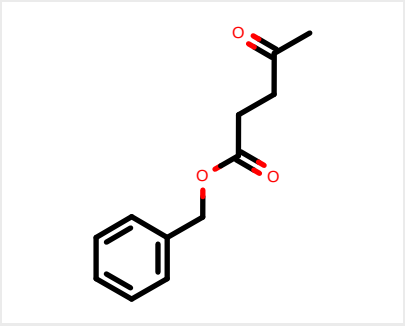
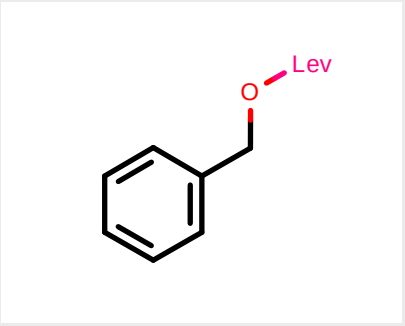
FPMP

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.FPMP*
	

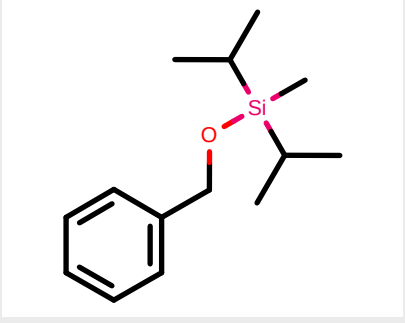
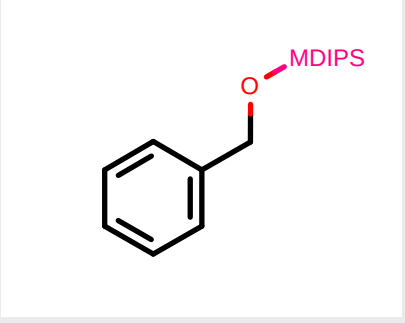
Im

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Im*
	

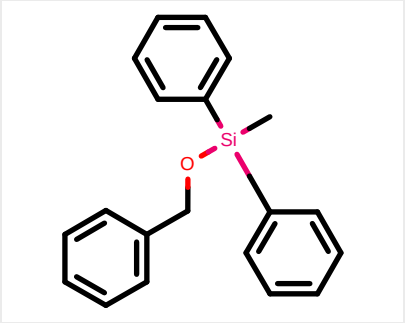
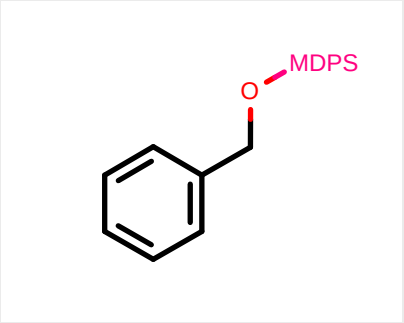
Lev

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Lev*
	

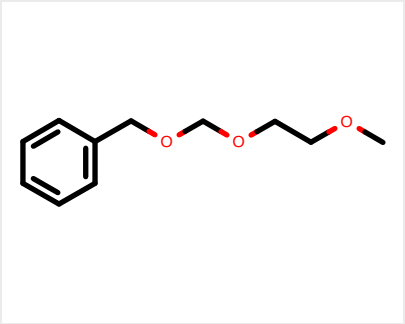
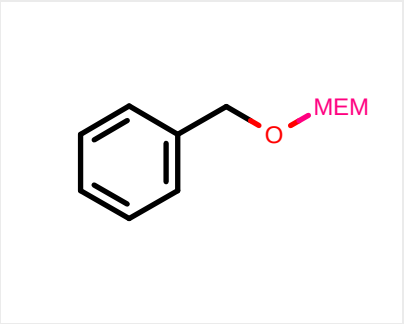
MDIPS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.MDIPS*
	

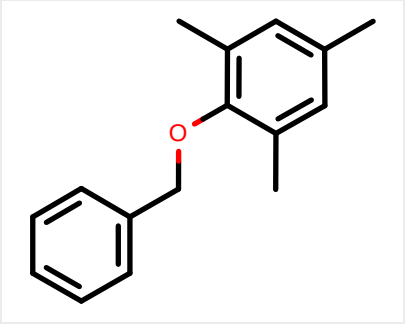
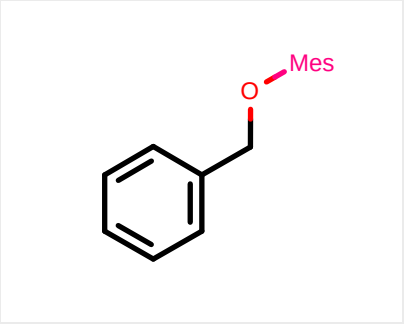
MDPS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.MDPS*
	

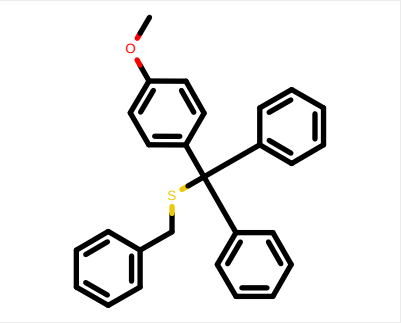
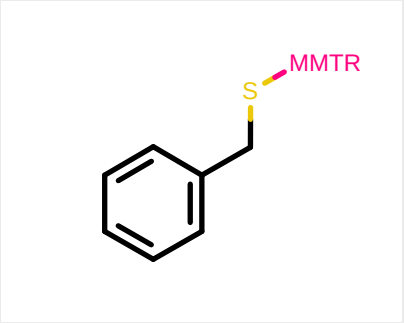
MEM

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.MEM*
	

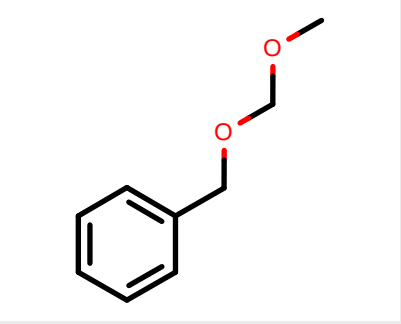
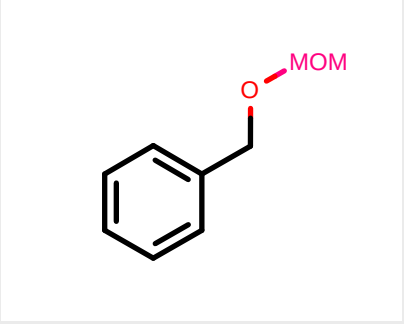
Mes

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Mes*
	

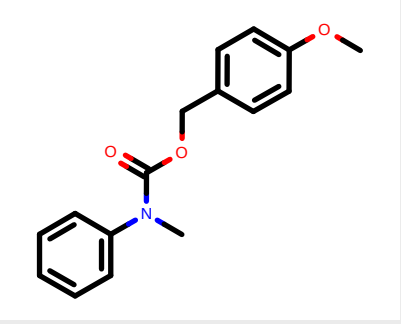
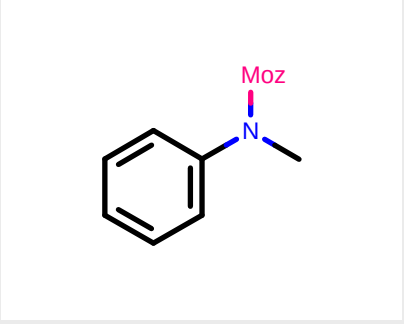
MMTR

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.MMTR*
	

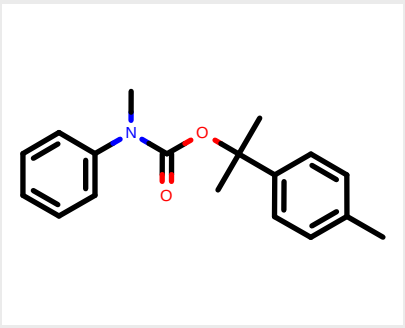
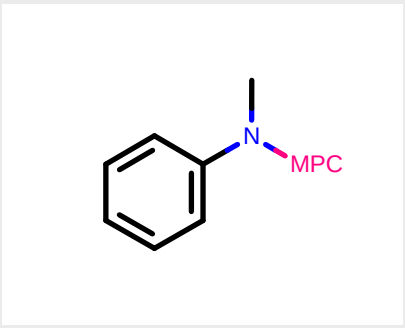
MOM

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.MOM*
	

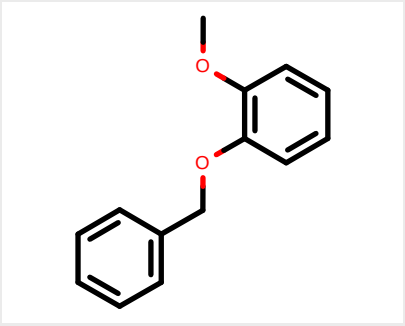
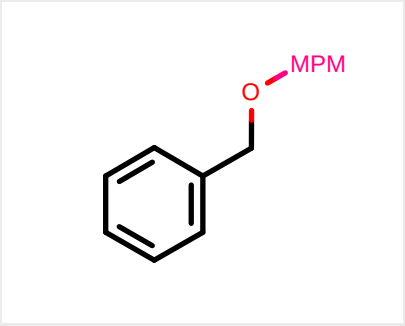
Moz

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Moz*
	

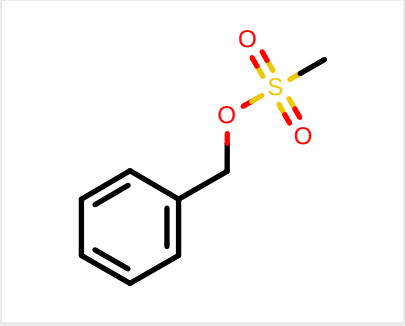
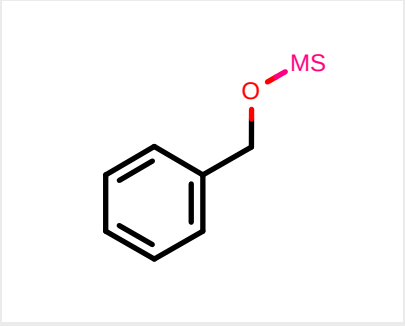
MPC

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.MPC*
	

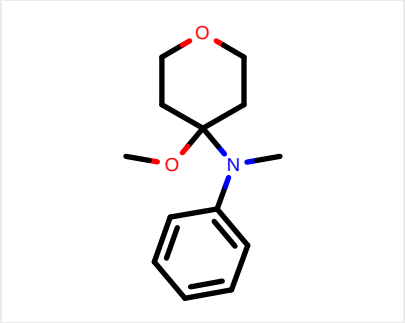
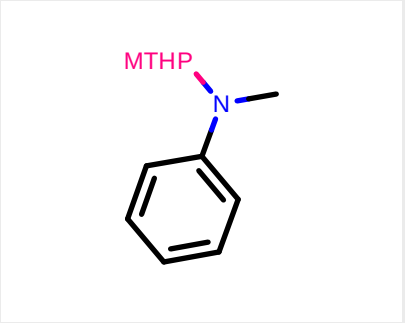
MPM

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.MPM*
	

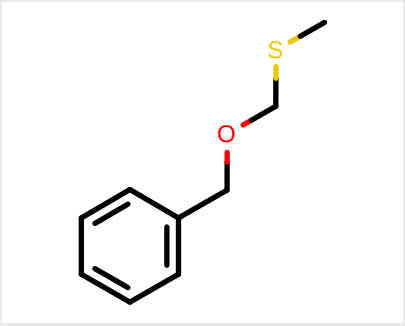
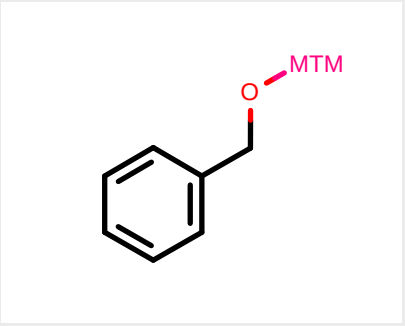
MS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.MS*
	

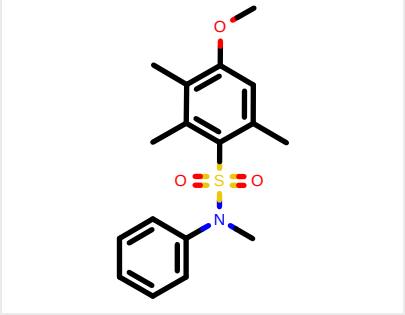
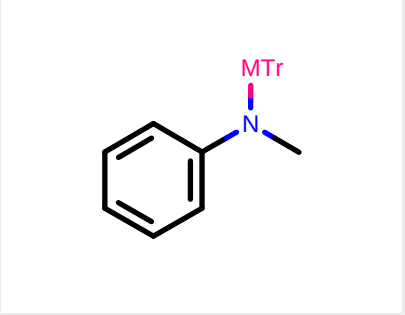
MTHP

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.MTHP*
	

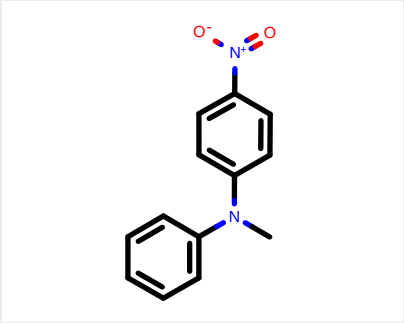
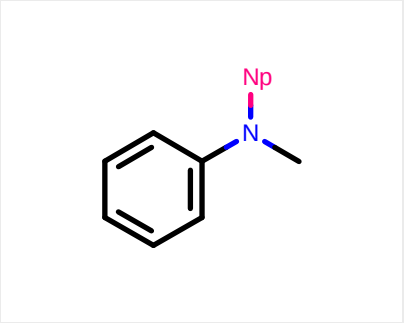
MTM

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.MTM*
	

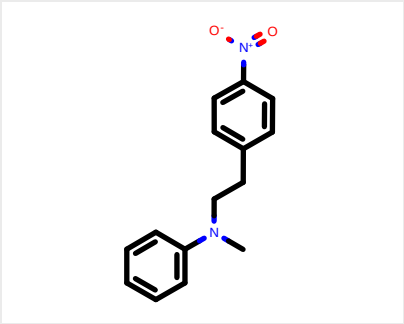
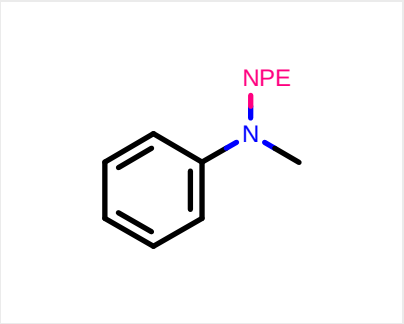
MTr

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.MTr*
	

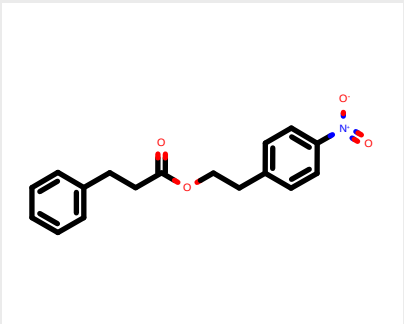
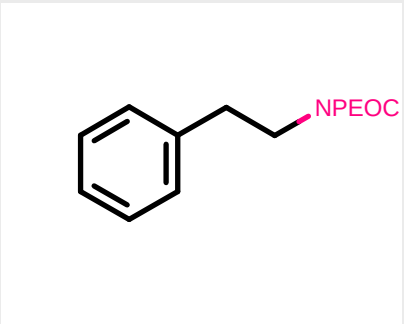
Np

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Np*
	

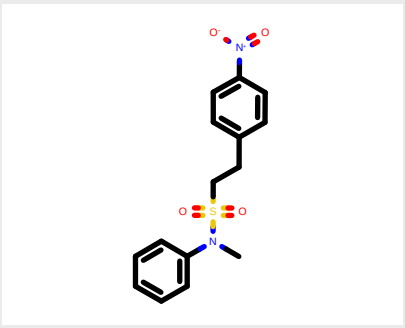
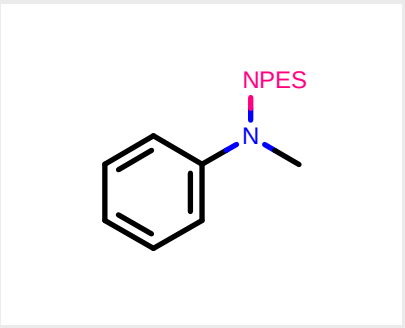
NPE

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.NPE*
	

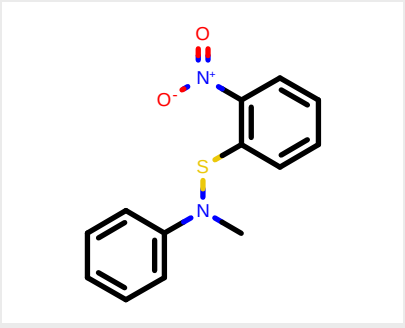
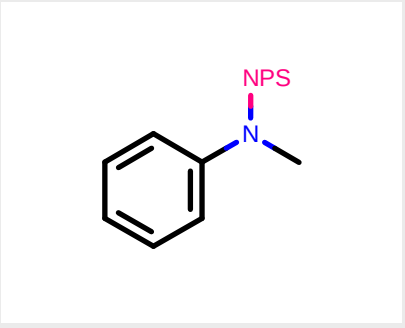
NPEOC

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.NPEOC*
	

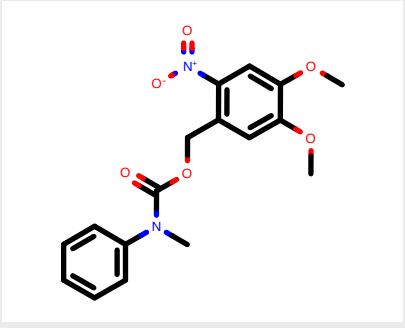
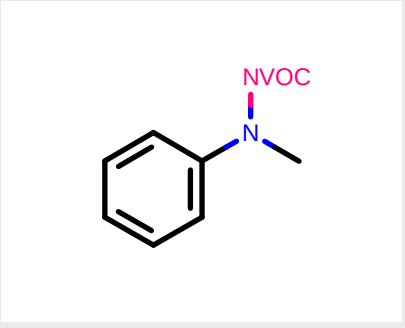
NPES

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.NPES*
	

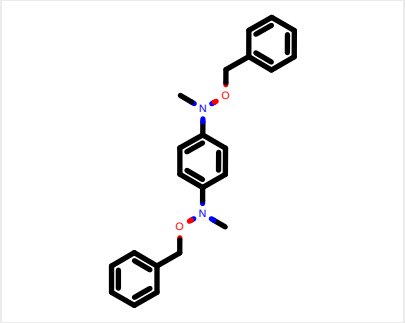
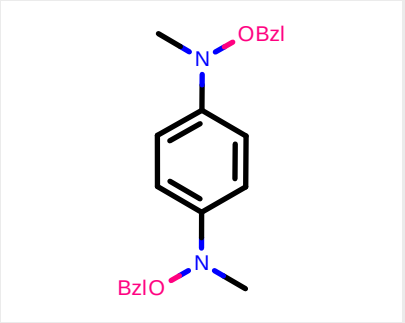
NPS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.NPS*
	

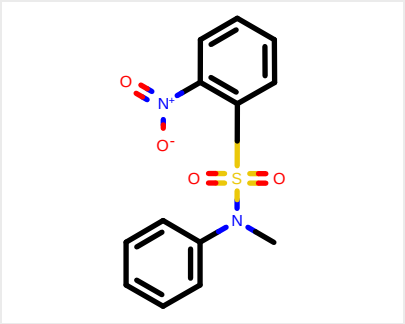
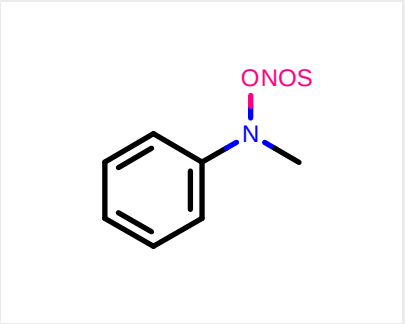
NVOC

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.NVOC*
	

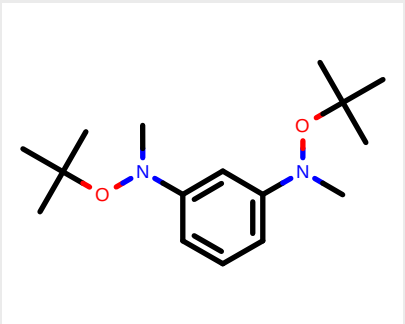
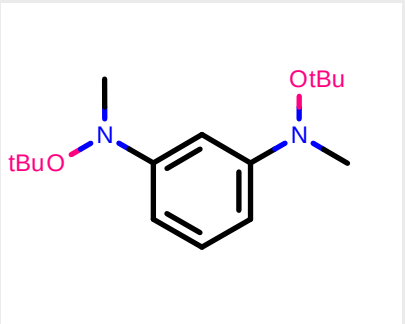
OBzl

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.OBzl*
	

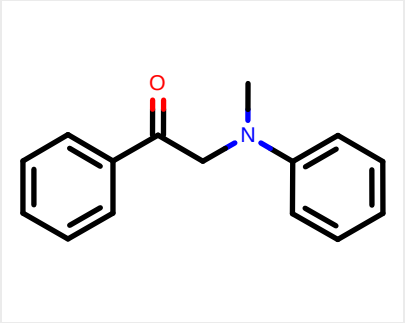
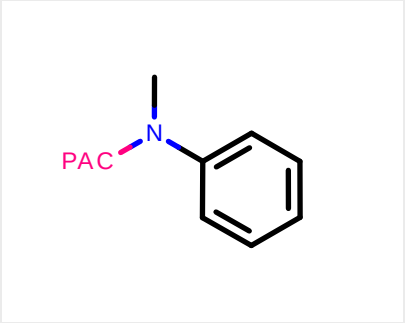
ONOS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.ONOS*
	

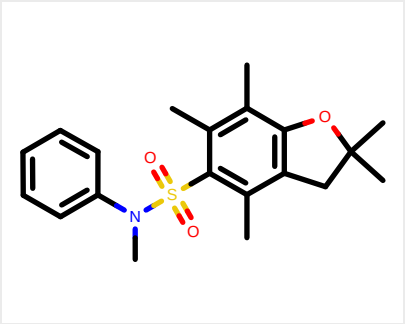
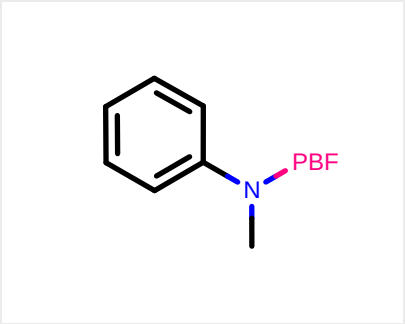
OtBu

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.OtBu*
	

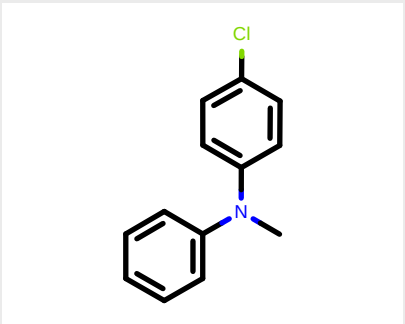
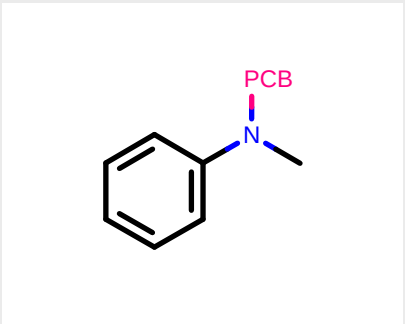
PAC

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.PAC*
	

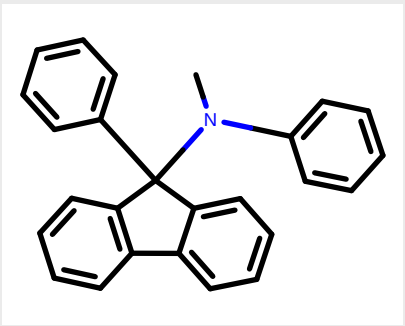
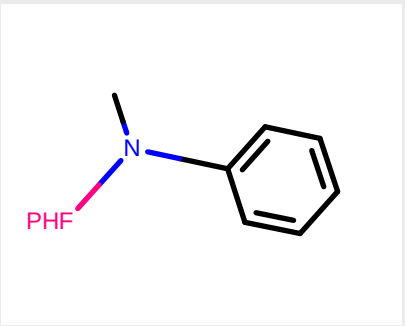
PBF

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.PBF*
	

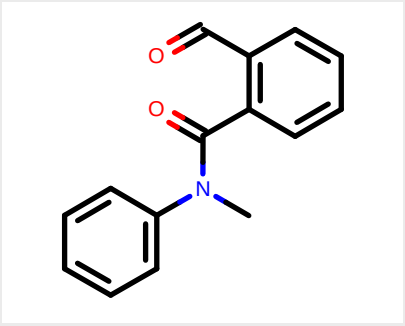
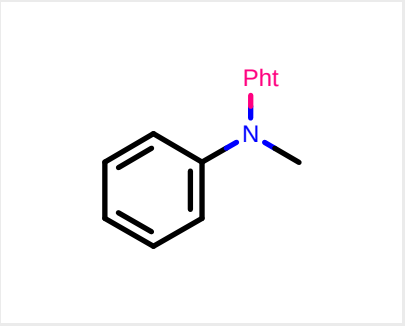
PCB

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.PCB*
	

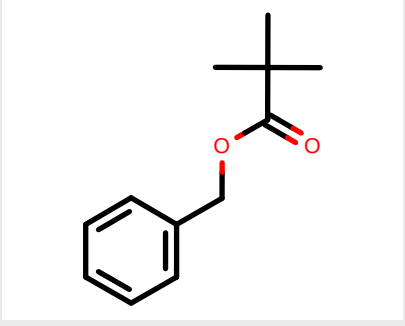
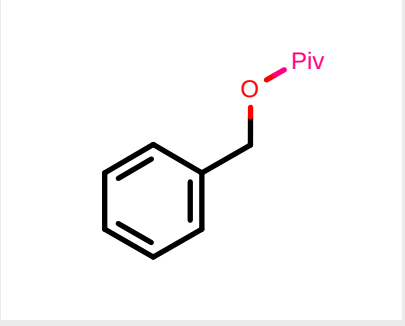
PHF

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.PHF*
	

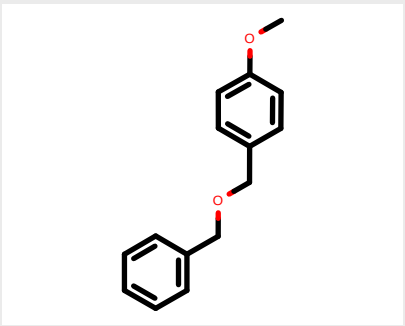
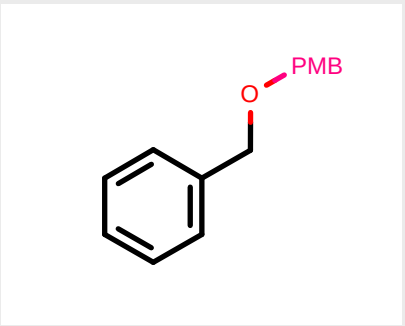
Pht

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Pht*
	

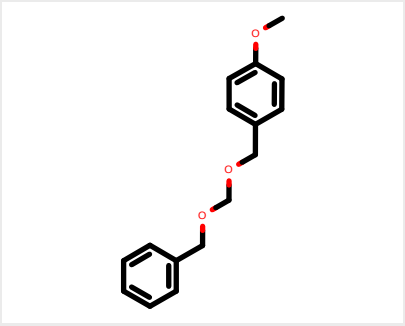
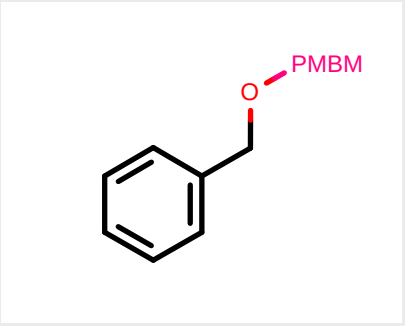
Piv

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Piv*
	

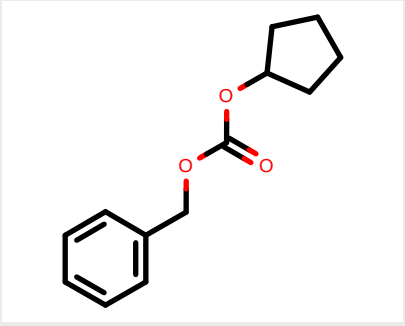
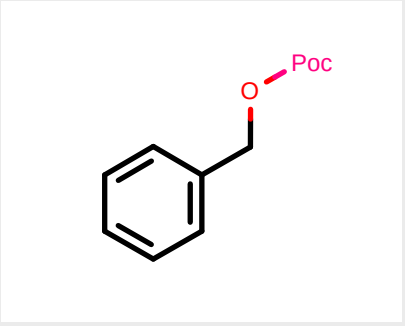
PMB

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.PMB*
	

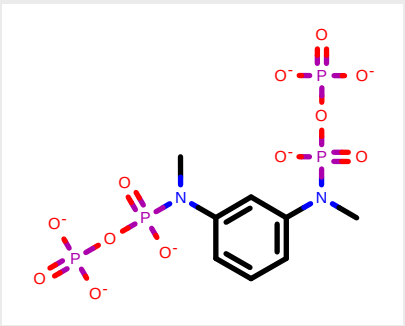
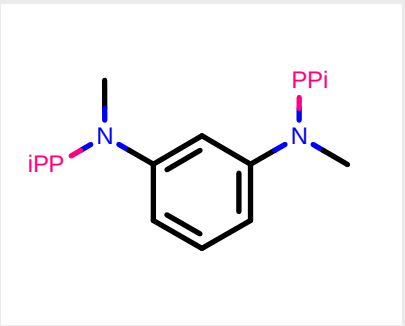
PMBM

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.PMBM*
	

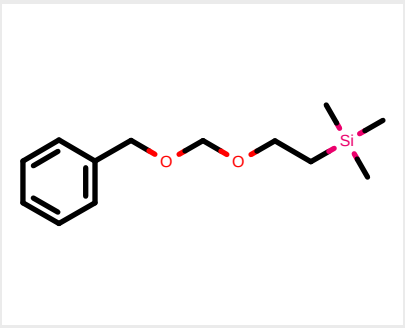
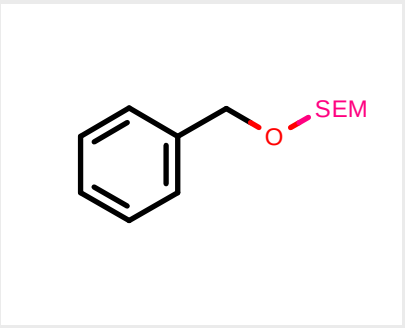
Poc

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Poc*
	

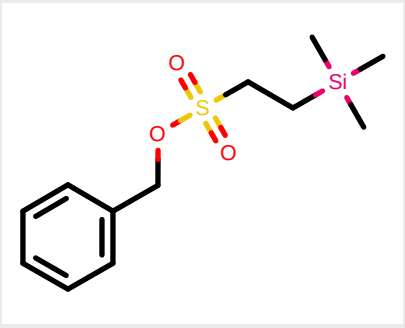
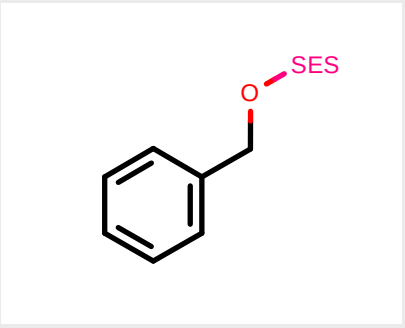
PPi

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.PPi*
	

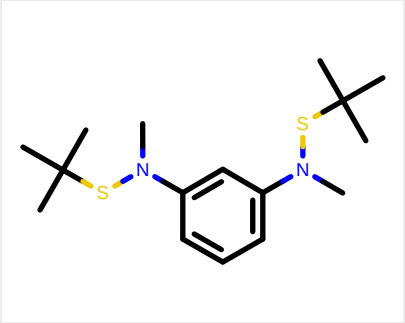
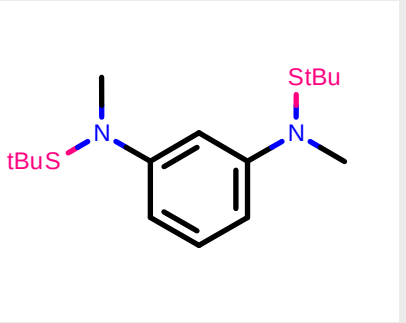
SEM

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.SEM*
	

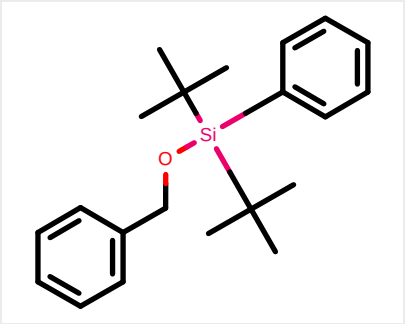
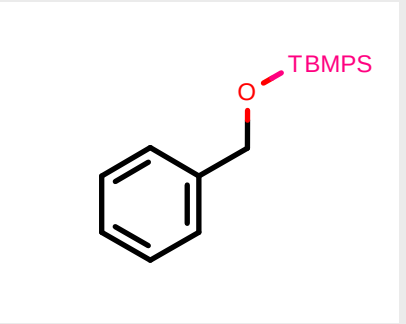
SES

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.SES*
	

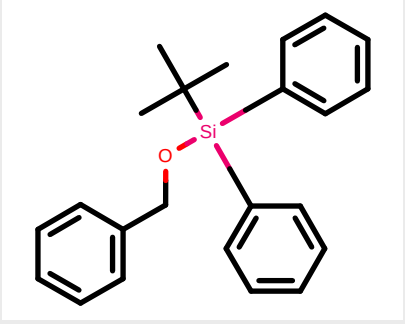
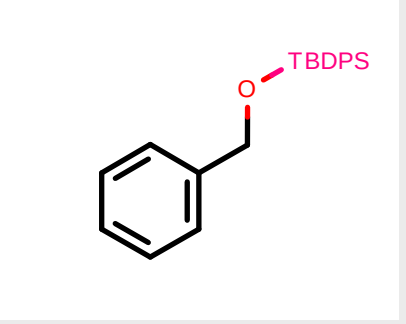
StBu

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.StBu*
	

TBMPS

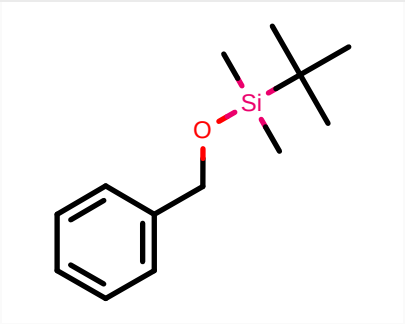
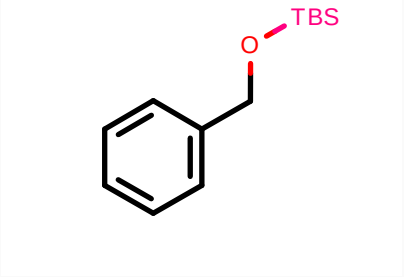
OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.TBMPS*
	

TBDPS

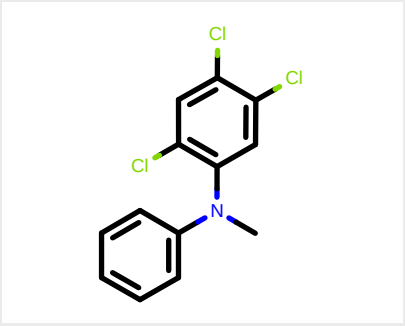
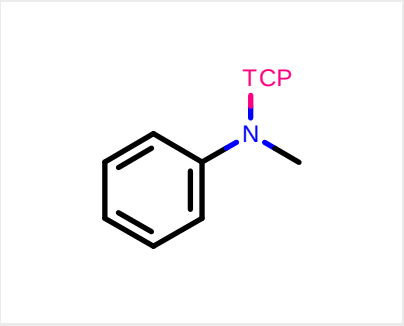
OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.TBDPS*
	

TBDMS

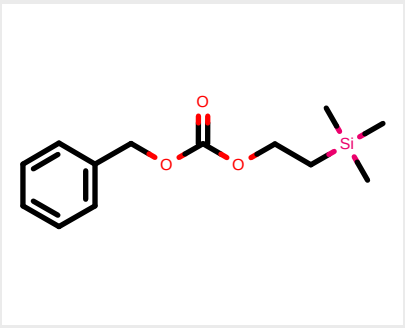
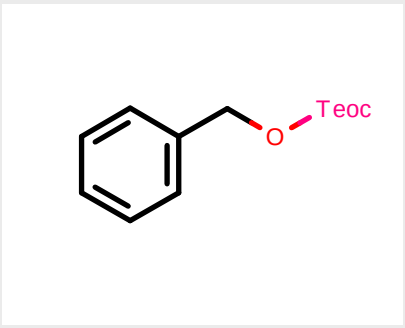
TBS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.TBDMS*
	OEProtectiveGroupStyle.TBS
	

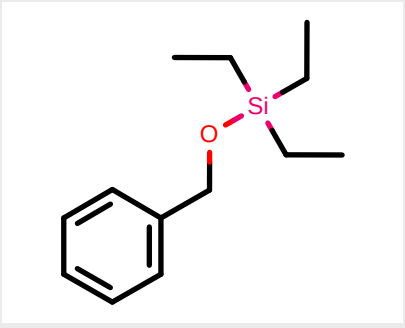
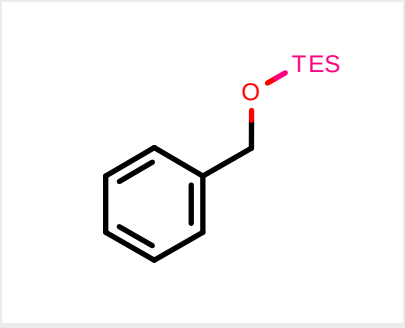
TCP

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.TCP*
	

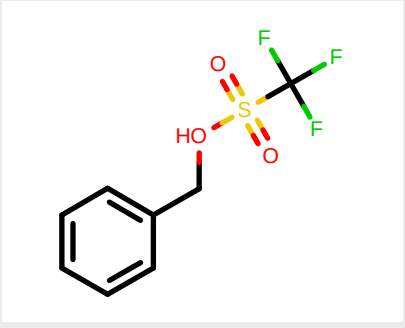
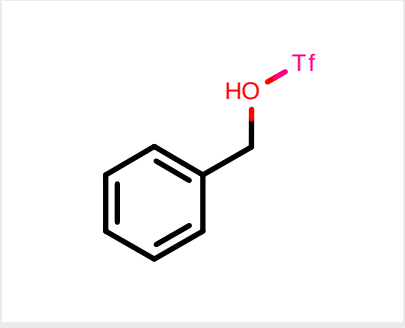
Teoc

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Teoc*
	

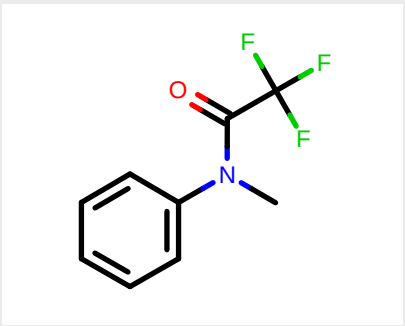
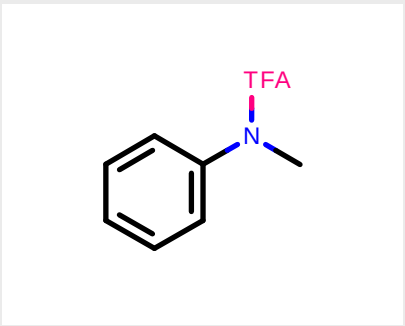
TES

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.TES*
	

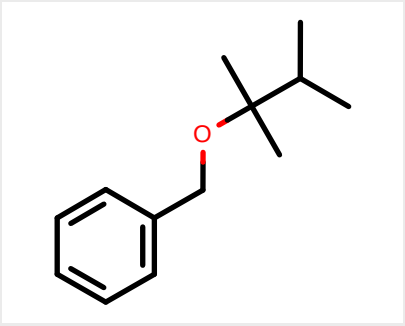
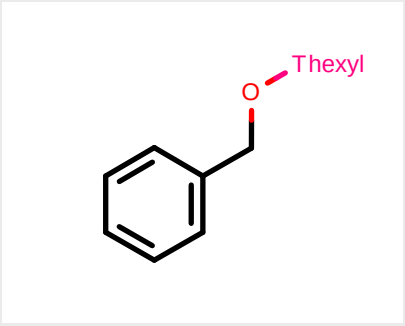
Tf

OEProtectiveGroupStyle.Tf	*OEProtectiveGroupStyle.Tf*
	

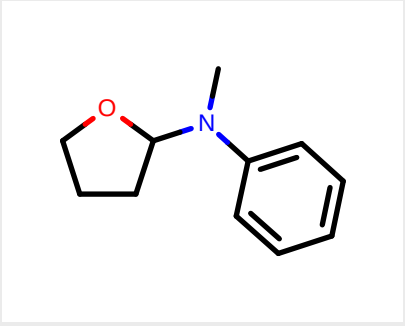
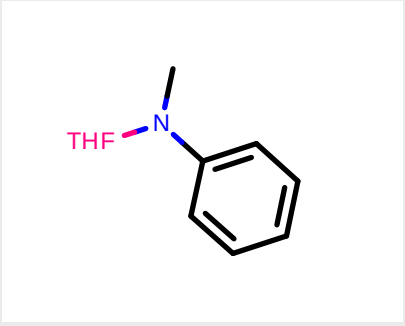
TFA

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.TFA*
	

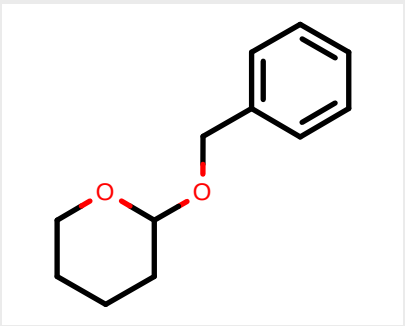
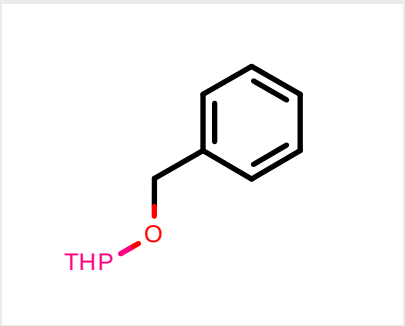
Thexyl

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Thexyl*
	

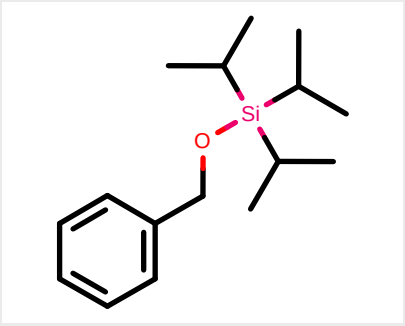
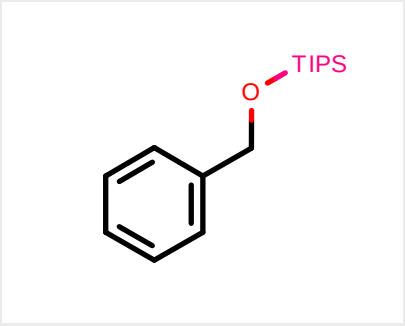
THF

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.THF*
	

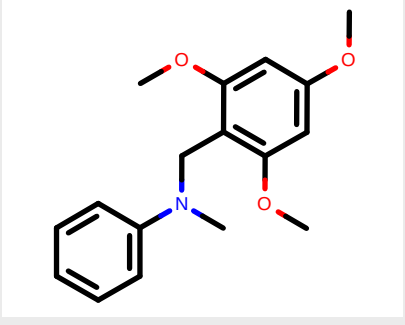
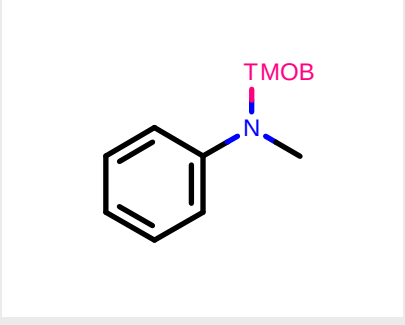
THP

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.THP*
	

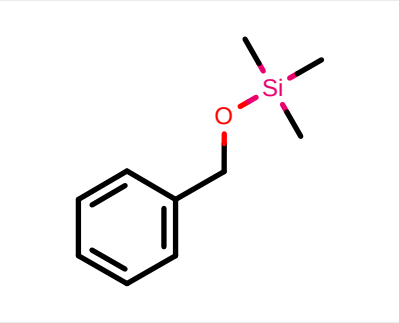
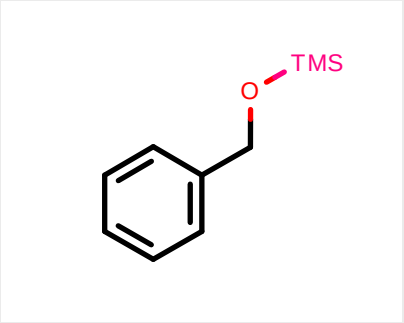
TIPS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.TIPS*
	

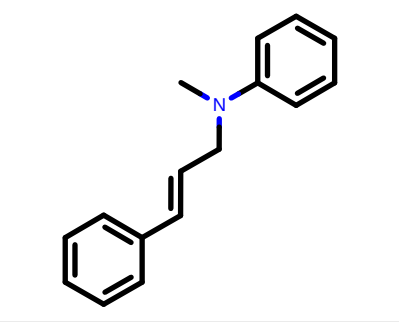
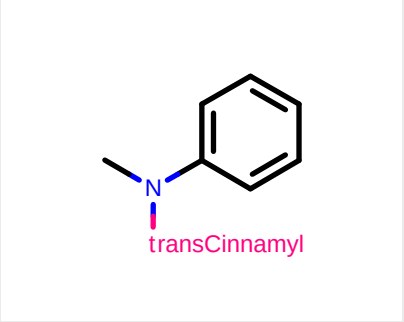
TMOB

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.TMOB*
	

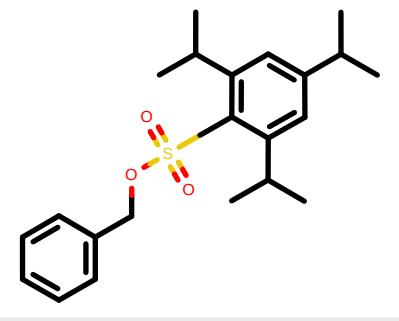
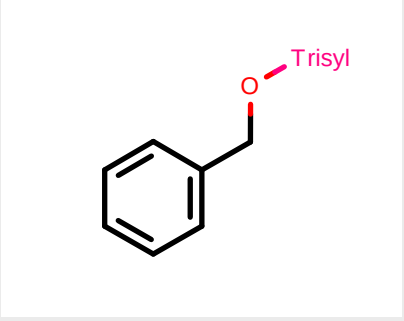
TMS

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.TMS*
	

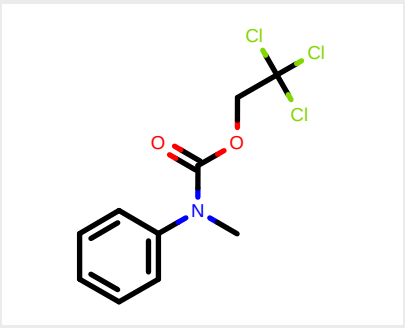
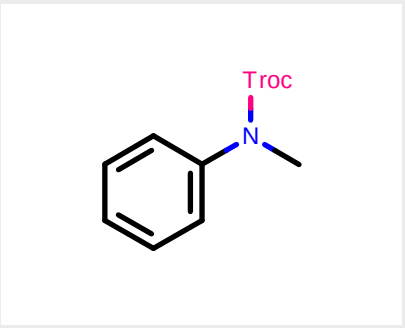
transCinnamyl

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.transCinnamyl*
	

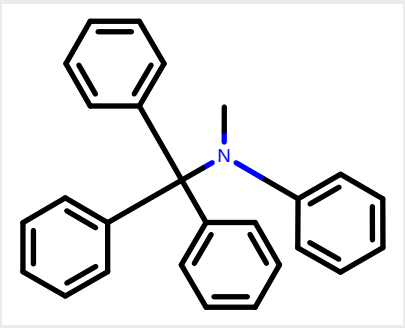
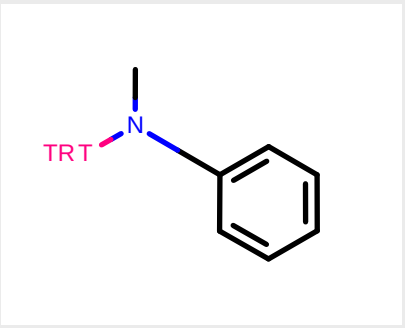
Trisyl

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Trisyl*
	

Troc

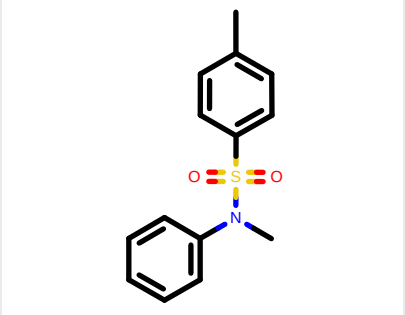
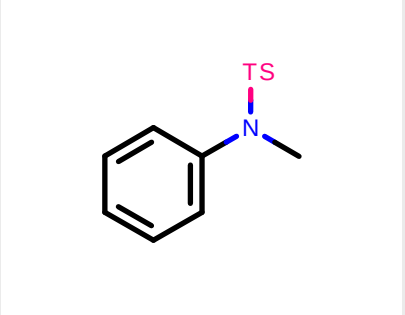
OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Troc*
	

TRT

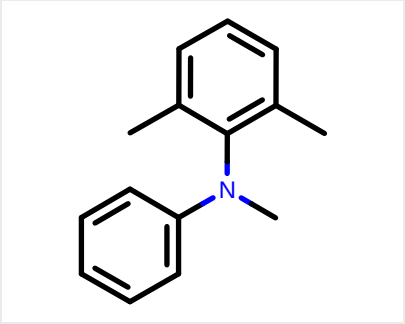
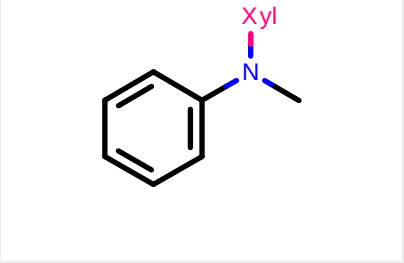
OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.TRT*
	

TS

Tos

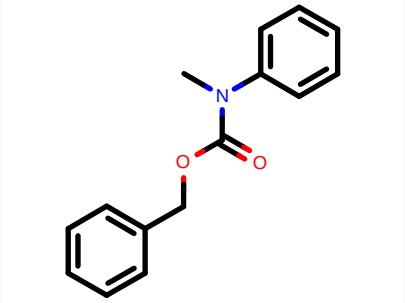
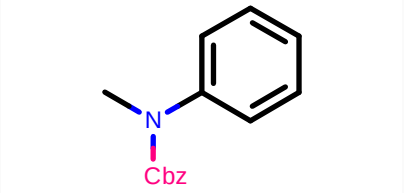
OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.TS*
	

Xyl

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Xyl*
	

Z

Cbz

OEProtectiveGroupStyle.Off	*OEProtectiveGroupStyle.Z*
	OEProtectiveGroupStyle.Cbz
	

4.2.42 OERadicalDisplayType

The *OERadicalDisplayType* namespace contains the following constants:

See also

- *OE2DAtomDisplay.GetRadical* method
- *OE2DAtomDisplay.SetRadical* method

Off

The atom is not a radical *i.e.* no information is rendered.

Diradical

Atom with two unpaired electrons. Either electron-paired (singlet state) or electron-unpaired (triplet state). See example in *Figure: Example of a diradical carbon depiction*.

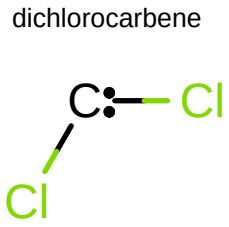


Fig. 324: **Example of a diradical carbon depiction**

Monoradical

Atom with one unpaired electron. See example in *Figure: Example of a monoradical carbon depiction*.

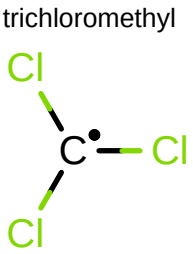


Fig. 325: **Example of a monoradical carbon depiction**

4.2.43 OEReportSetup

This namespace encodes symbolic constants used as bit-masks to indicate which image report layout parameters are being created when invoking the `OEConfigureReportOptions` function.

See also

- *OEReport* class
- *OEReportOptions* class

All

The combination of following constants:

- *OEReportSetup.NumColsPerPage*
- *OEReportSetup.NumRowsPerPage*
- *OEReportSetup.PageHeight*
- *OEReportSetup.PageOrientation*
- *OEReportSetup.PageSize*
- *OEReportSetup.PageWidth*

NumColsPerPage

Passing this constant to the *OEConfigureReportOptions* function result in generating the following default interface to configure the number of columns per page.

```
Contents of parameter -colsperpage
    Aliases : -cols
    Type : int
    Allow list : false
    Default : 2
    Simple : true
    Required : false
    Legal ranges :
        10 to 1
    Brief : Number of columns per page
```

See also

- *OEReportOptions.NumColsPerPage* method

NumRowsPerPage

Passing this constant to the *OEConfigureReportOptions* function result in generating the following default interface to configure the number of rows per page.

```
Contents of parameter -rowsperpage
    Aliases : -rows
    Type : int
    Allow list : false
    Default : 3
    Simple : true
    Required : false
    Legal ranges :
        10 to 1
    Brief : Number of rows per page
```

See also

- *OEReportOptions.NumRowsPerPage* method

PageHeight

Passing this constant to the *OEConfigureReportOptions* function result in generating the following default interface to configure the page height of the report.

```
Contents of parameter -pageheight
    Aliases : -h
    Type : double
    Allow list : false
    Default : (parameter does not have a default)
    Simple : true
    Required : false
    Legal ranges :
        2000.0 to 100.0
    Brief : Page height
```

See also

- *OEReportOptions.SetPageHeight* method

PageOrientation

Passing this constant to the *OEConfigureReportOptions* function result in generating the following default interface to configure the page orientation of the report.

```
Contents of parameter -pageorientation
    Aliases : -pori
    Type : string
    Allow list : false
    Default : Portrait
    Simple : true
    Required : false
    Legal values : Portrait Landscape
    Brief : Page orientation
```

See also

- *OEReportOptions.SetPageOrientation* method
- *OEPageOrientation* namespace

PageSize

Passing this constant to the *OEConfigureReportOptions* function result in generating the following default interface to configure the page size of the report.

```
Contents of parameter -pagesize
    Aliases : -psize
    Type : string
    Allow list : false
    Default : US_Letter
    Simple : true
    Required : false
    Legal values : ISO_A4 US_Letter US_Legal Custom
    Brief : Page size
```

See also

- *OEReportOptions.SetPageSize* method
- *OEPageSize* namespace

PageWidth

Passing this constant to the *OEConfigureReportOptions* function result in generating the following default interface to configure the page width of the report.

```
Contents of parameter -pagewidth
    Aliases : -w
    Type : double
    Allow list : false
    Default : (parameter does not have a default)
    Simple : true
    Required : false
    Legal ranges :
        2000.0 to 100.0
    Brief : Page width
```

See also

- *OEReportOptions.SetPageWidth* method

4.2.44 OEScale

This namespace contains constants that determine the scaling factor of a molecule when positioned in an image.

See also

- *OE2DMolDisplayOptions.GetScale*
- *OE2DMolDisplayOptions.SetScale*

- *OE2DMolDisplayOptions.SetDimensions*

The *OEScale* namespace contains the following constants:

Default

Specifies the default scaling of a molecule.

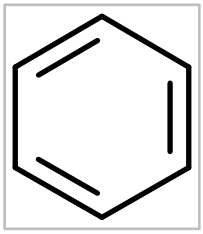


Fig. 326: **Default scaling of benzene**

AutoScale

Auto scaling means that a depicted molecule is scaled to maximally fit the given image dimensions.

See also

- *Controlling the Size of Depicted Molecules* section

4.2.45 OEStipple

This namespace contains constants representing various stipple patterns.

See also

- *OEPen.GetLineStipple* method
- *OEPen.SetLineStipple* method
- *OEPen.GetStippleFactor* method
- *OEPen.SetStippleFactor* method

The *OEStipple* namespace contains the following constants:

Default

The default line stipple style is *OEStipple.Solid*.

Dot

See example in *Figure: Example of line drawing with 'Dot' stipple style*.

Fig. 327: **Example of line drawing with 'Dot' stipple style**

DotDash

See example in *Figure: Example of line drawing with 'DotDash' stipple style*.

Fig. 328: **Example of line drawing with 'DotDash' stipple style**

DotDashDash

See example in *Figure: Example of line drawing with 'DotDashDash' stipple style*.

Fig. 329: **Example of line drawing with 'DotDashDash' stipple style**

LongDash

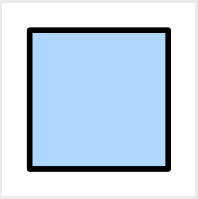
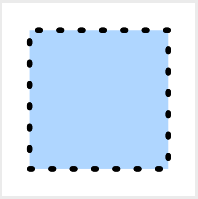
See example in *Figure: Example of line drawing with 'LongDash' stipple style*.

Fig. 330: **Example of line drawing with 'LongDash' stipple style**

NoLine

See example in *Table: Example of drawing rectangles with various stipple styles*.

Table 44: **Example of drawing rectangles with various stipple styles**

OEStipple.Solid	*OEStipple.Dot*	*OEStipple.NoLine*
		

ShortDash

See example in *Figure: Example of line drawing with 'ShortDash' stipple style*.

Fig. 331: **Example of line drawing with 'ShortDash' stipple style**

Solid

See example in *Figure: Example of line drawing with 'Solid' stipple style*.

Fig. 332: **Example of line drawing with 'Solid' stipple style**

4.2.46 OESuperAtomStyle

This namespace contains constants representing the various super atom styles supported by **OEDepict TK**. Super atoms are representing specific functional groups that are contracted into a single label.

See also

- *OE2DMolDisplayOptions.GetSuperAtomStyle* method
- *OE2DMolDisplayOptions.SetSuperAtomStyle* method
- *OEProtectiveGroupStyle* namespace

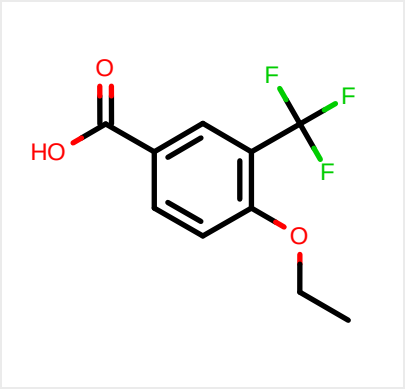
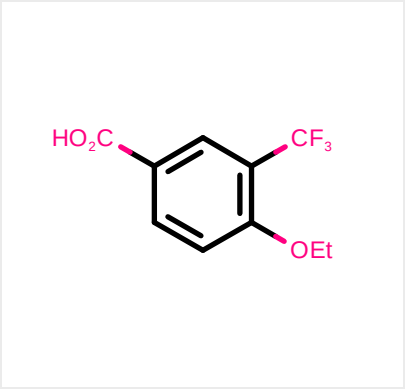
The *OESuperAtomStyle* namespace contains the following constants:

Default

The default super atom display style is *OESuperAtomStyle.Off*.

All

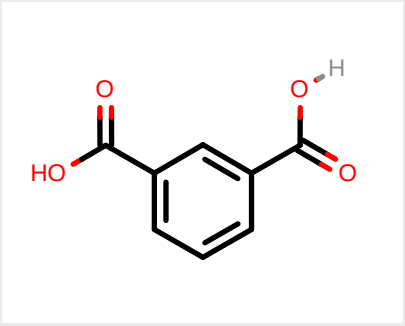
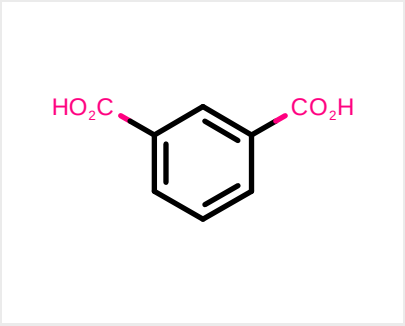
The combination of all super atom flags.

OESuperAtomStyle.Off	*OESuperAtomStyle.All*
	

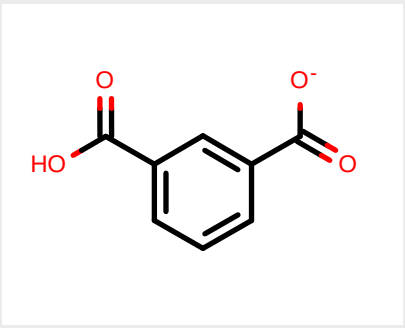
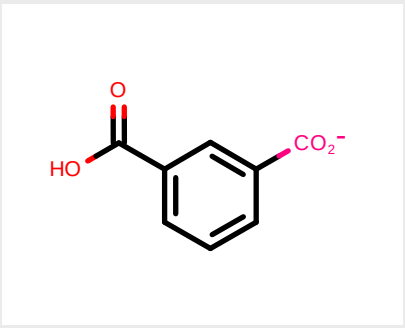
Off

No structural abbreviation is performed.

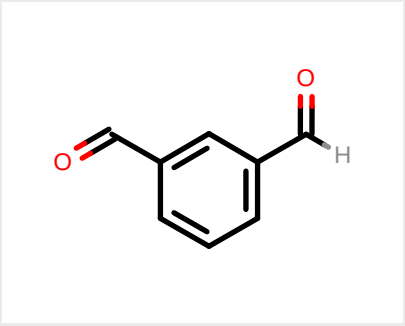
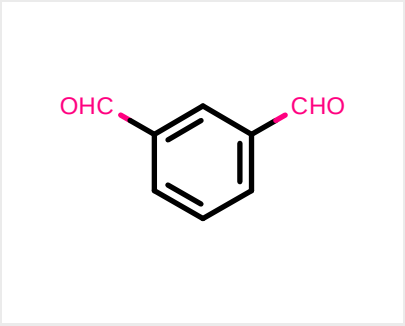
COOH

OESuperAtomStyle.Off	*OESuperAtomStyle.COOH*
	

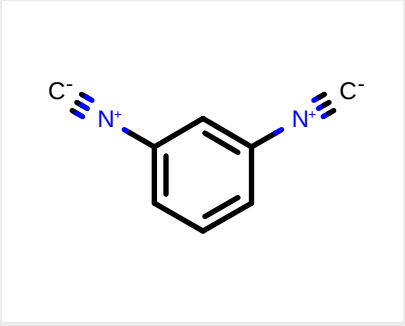
COO_

OESuperAtomStyle.Off	*OESuperAtomStyle.COO_*
	

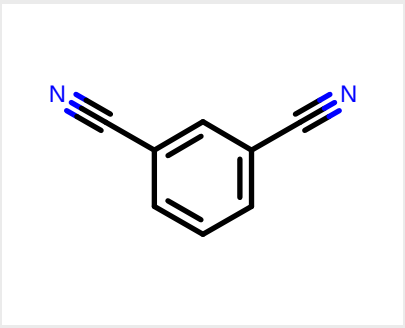
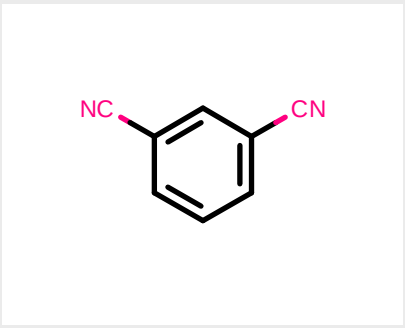
CHO

OESuperAtomStyle.Off	*OESuperAtomStyle.CHO*
	

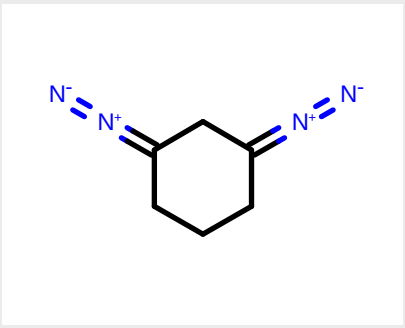
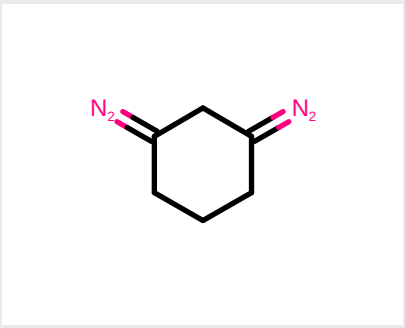
NC

OESuperAtomStyle.Off	*OESuperAtomStyle.NC*
	

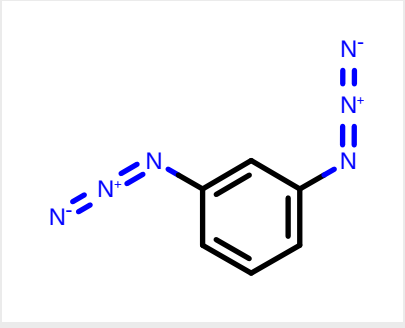
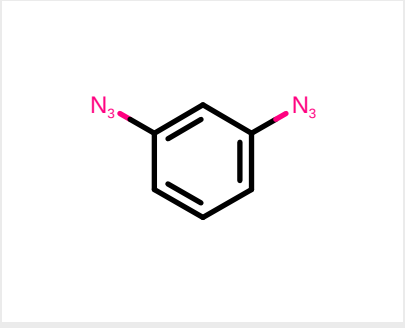
CN

OESuperAtomStyle.Off	*OESuperAtomStyle.CN*
	

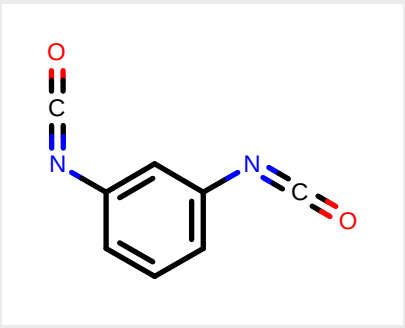
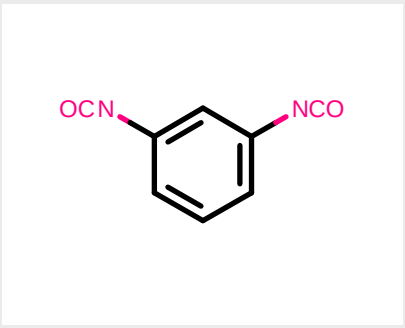
N2

OESuperAtomStyle.Off	*OESuperAtomStyle.N2*
	

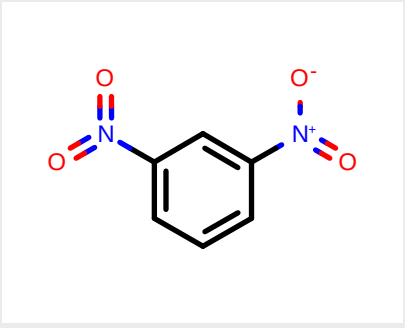
N3

OESuperAtomStyle.Off	*OESuperAtomStyle.N3*
	

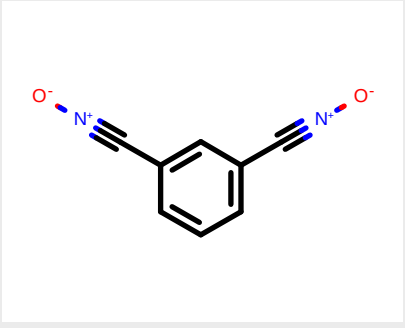
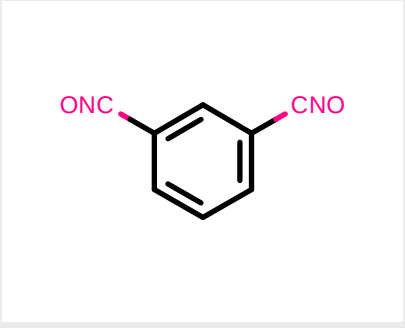
NCO

OESuperAtomStyle.Off	*OESuperAtomStyle.NCO*
	

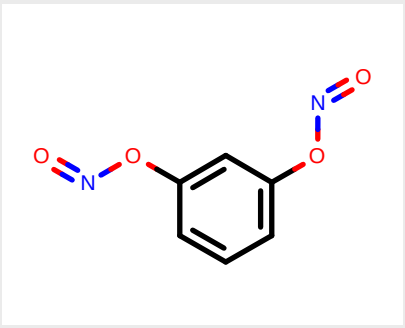
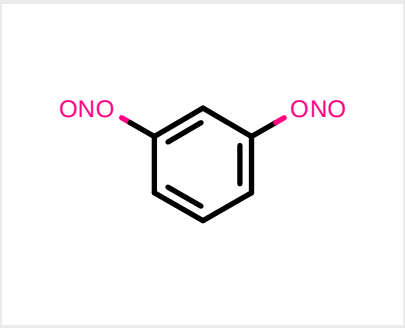
NO2

OESuperAtomStyle.Off	*OESuperAtomStyle.NO2*
	

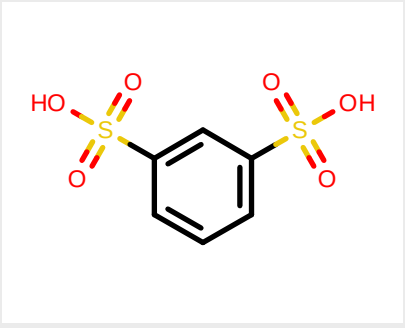
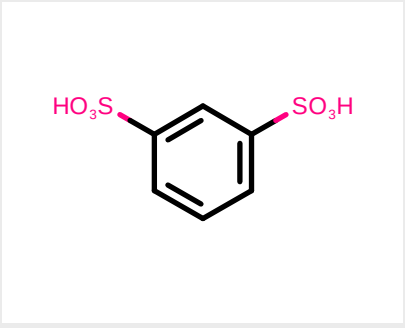
CNO

OESuperAtomStyle.Off	*OESuperAtomStyle.CNO*
	

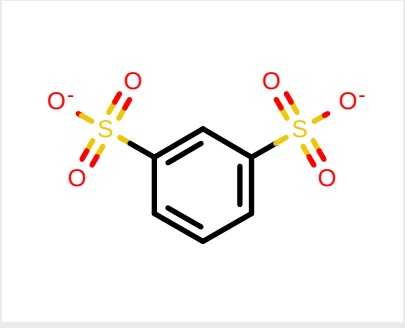
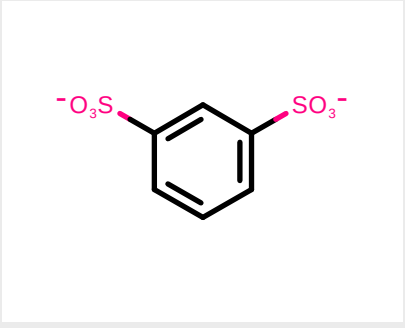
ONO

OESuperAtomStyle.Off	*OESuperAtomStyle.ONO*
	

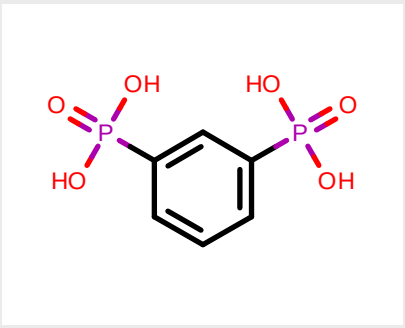
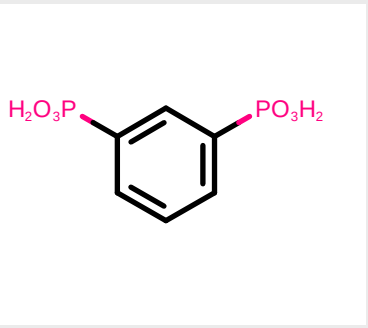
SO3H

OESuperAtomStyle.Off	*OESuperAtomStyle.SO3H*
	

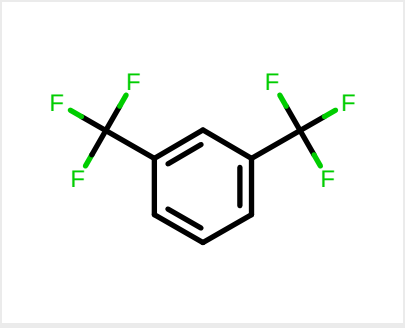
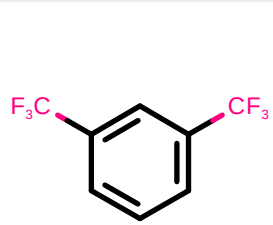
SO3_

OESuperAtomStyle.Off	*OESuperAtomStyle.SO3_*
	

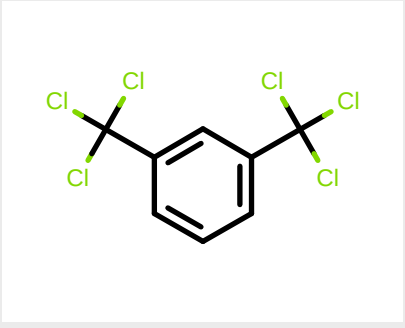
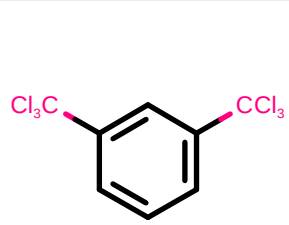
PO3H2

OESuperAtomStyle.Off	*OESuperAtomStyle.PO3H2*
	

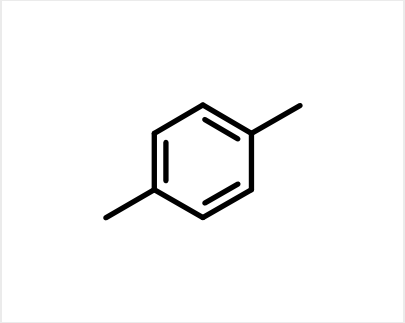
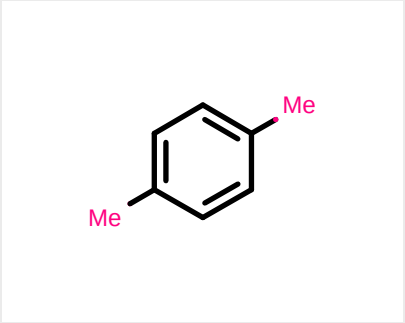
CF3

OESuperAtomStyle.Off	*OESuperAtomStyle.CF3*
	

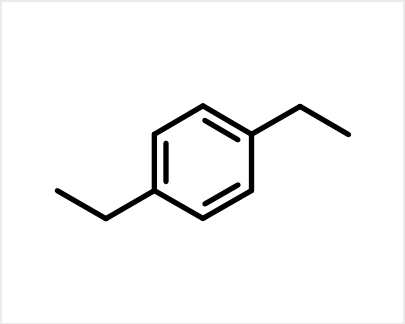
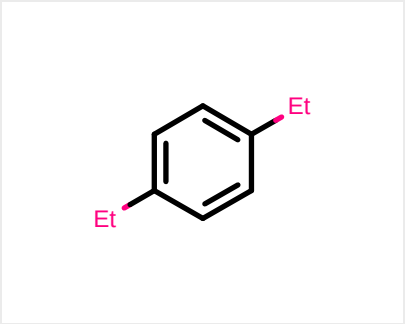
CCl3

OESuperAtomStyle.Off	*OESuperAtomStyle.CCl3*
	

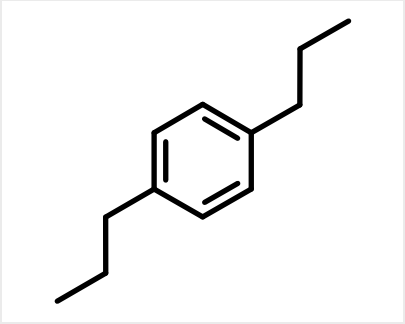
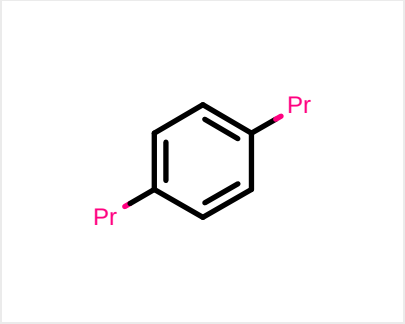
Me

OESuperAtomStyle.Off	*OESuperAtomStyle.Me*
	

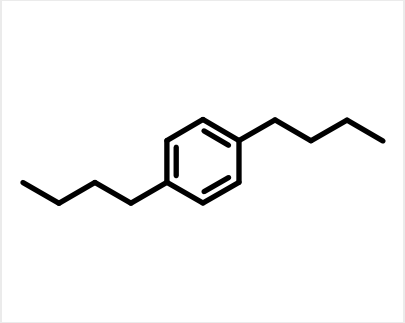
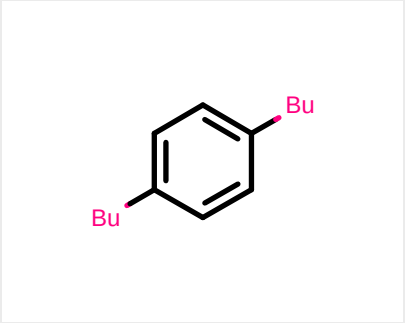
Et

OESuperAtomStyle.Off	*OESuperAtomStyle.Et*
	

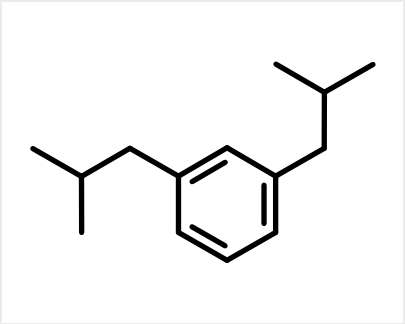
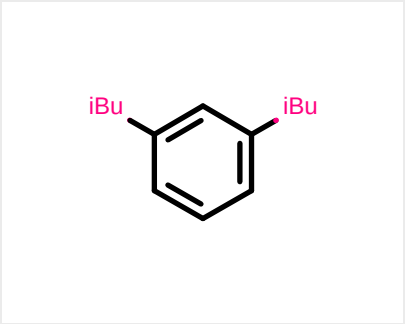
Pr

OESuperAtomStyle.Off	*OESuperAtomStyle.Pr*
	

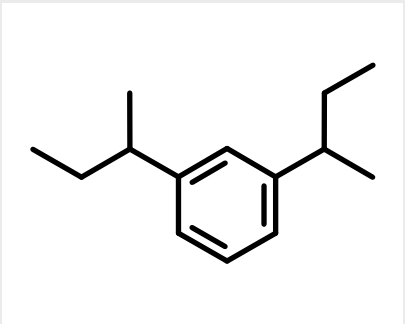
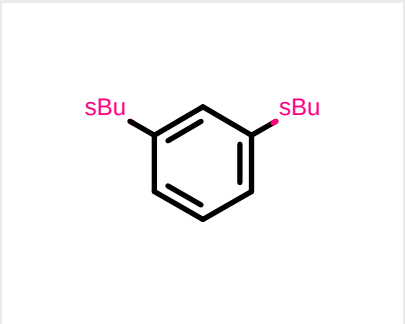
Bu

OESuperAtomStyle.Off	*OESuperAtomStyle.Bu*
	

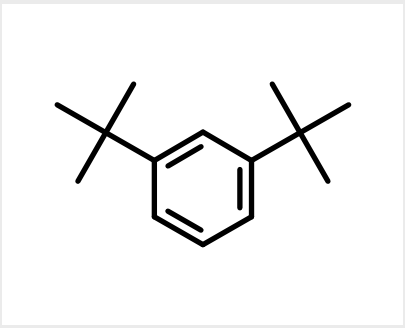
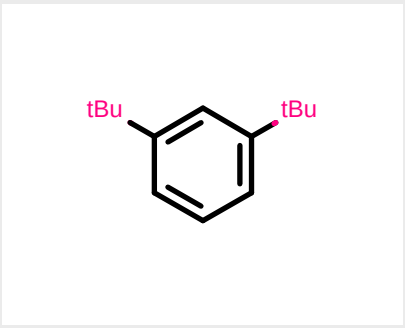
iBu

OESuperAtomStyle.Off	*OESuperAtomStyle.iBu*
	

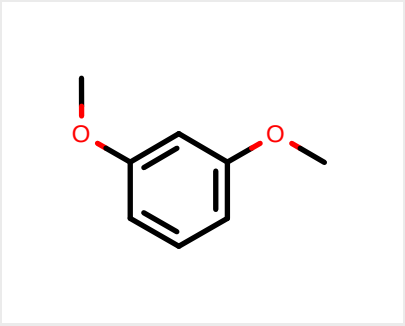
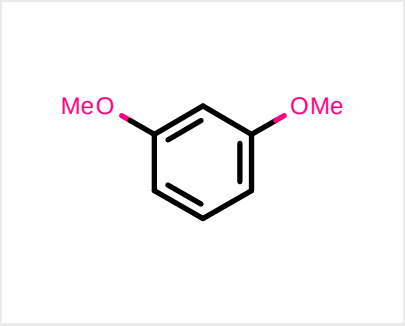
sBu

OESuperAtomStyle.Off	*OESuperAtomStyle.sBu*
	

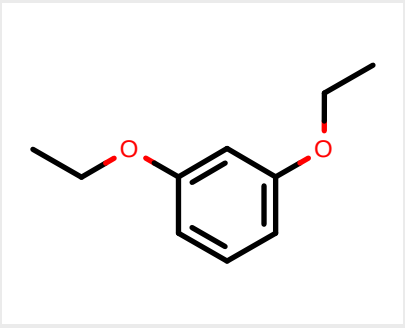
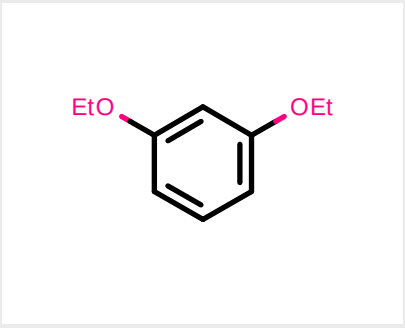
tBu

OESuperAtomStyle.Off	*OESuperAtomStyle.tBu*
	

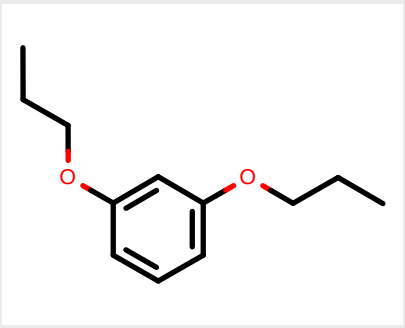
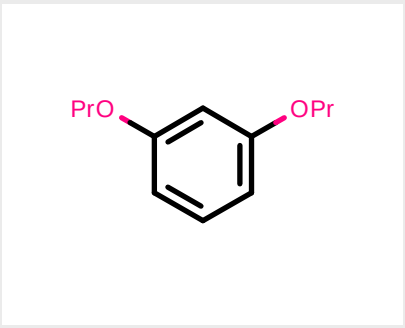
OMe

OESuperAtomStyle.Off	*OESuperAtomStyle.OMe*
	

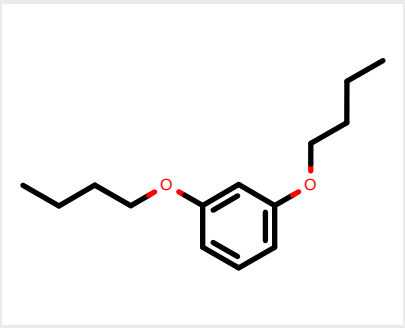
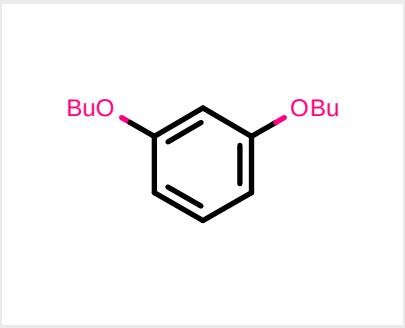
OEt

OESuperAtomStyle.Off	*OESuperAtomStyle.OEt*
	

OPr

OESuperAtomStyle.Off	*OESuperAtomStyle.OPr*
	

OBu

OESuperAtomStyle.Off	*OESuperAtomStyle.OBu*
	

Oxygen

The combination of following super atom flags:

- *OESuperAtomStyle.COOH*
- *OESuperAtomStyle.COO_*
- *OESuperAtomStyle.CHO*

Nitrogen

The combination of the following super atom flags:

- *OESuperAtomStyle.NC*
- *OESuperAtomStyle.CN*
- *OESuperAtomStyle.N2*
- *OESuperAtomStyle.N3*

OxygenAndNitrogen

The combination of the following super atom flags:

- *OESuperAtomStyle.NCO*
- *OESuperAtomStyle.NO2*
- *OESuperAtomStyle.CNO*
- *OESuperAtomStyle.ONO*

Sulfur

The combination of the following super atom flags:

- *OESuperAtomStyle.SO3H*
- *OESuperAtomStyle.SO3_*

Phosphorus

The combination of the following super atom flags:

- *OESuperAtomStyle.PO3H2*

Halide

The combination of the following super atom flags:

- *OESuperAtomStyle.CF3*
- *OESuperAtomStyle.CCl3*

Carbon

The combination of the following super atom flags:

- *OESuperAtomStyle.Me*
- *OESuperAtomStyle.Et*
- *OESuperAtomStyle.Pr*
- *OESuperAtomStyle.Bu*
- *OESuperAtomStyle.iBu*
- *OESuperAtomStyle.sBu*
- *OESuperAtomStyle.tBu*

Ether

The combination of the following super atom flags:

- *OESuperAtomStyle.OMe*
- *OESuperAtomStyle.OEt*
- *OESuperAtomStyle.OPr*
- *OESuperAtomStyle.OBu*

4.2.47 OETitleLocation

This namespace contains constants representing the position of the molecule title:

See also

- *OE2DMolDisplayOptions.GetTitleLocation*
- *OE2DMolDisplayOptions.SetTitleLocation*

The *OETitleLocation* namespace contains the following constants:

Default

The default title location is *OETitleLocation.Top*.

Bottom

See example in *Figure: Example of displaying molecule title at the bottom*.

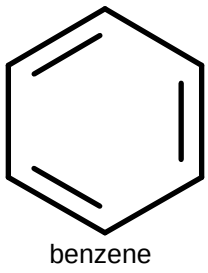


Fig. 333: **Example of displaying molecule title at the bottom**

Hidden

See example in *Figure: Example of displaying molecule with no title*.

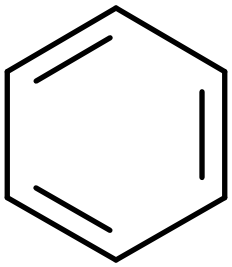


Fig. 334: **Example of displaying molecule with no title**

Top

See example in *Figure: Example of displaying molecule title at the top*.

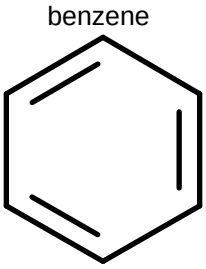


Fig. 335: **Example of displaying molecule title at the top**

4.3 OEDepict Functions

4.3.1 OEAddBuiltInHiddenSVGClass

```
const OESVGClass* OEAddBuiltInHiddenSVGClass(OEImageBase& image)
```

Adds an *OESVGClass* to the given image that can be used to mark up a group of SVG elements to be hidden.

See also

- *OESVGClass* class
- *OEImage* class
- *OEAddBuiltInVisibleSVGClass* function
- *OEGetBuiltInHiddenSVGClass* function
- *OEGetBuiltInVisibleSVGClass* function

4.3.2 OEAddBuiltInVisibleSVGClass

```
const OESVGClass* OEAddBuiltInVisibleSVGClass(OEImageBase& image)
```

Adds an *OESVGClass* to the given image that can be used to mark up a group of SVG elements to be visible.

See also

- *OESVGClass* class
- *OEImage* class
- *OEAddBuiltInHiddenSVGClass* function
- *OEGetBuiltInHiddenSVGClass* function
- *OEGetBuiltInVisibleSVGClass* function

4.3.3 OEAddDepictionHydrogens

```
bool OEAddDepictionHydrogens(OEChem::OEMolBase& mol)
```

Adds explicit "depiction" hydrogens to the specified OEMolBase. Depiction hydrogens are hydrogens that need to be explicitly drawn in a 2D depiction to faithfully represent tetrahedral atom stereochemistry or cis/trans bond stereochemistry.

The *OEAddDepictionHydrogens* function identifies all implicit depiction hydrogens (using the *OEHasDepictionHydrogens* function), and converts them into explicit hydrogens using the `OEAddExplicitHydrogens` function. This function returns true if any depiction hydrogens were converted into explicit atoms.

See also

- *OEHasDepictionHydrogens* function
- *OESuppressNonDepictionHydrogens* function

4.3.4 OEAddHighlighting

Highlights atoms and/or bonds of the displayed molecule.

Table 45: **The overloaded versions of the OEAddHighlighting function using OEHighlightBase**

Link	Description
OEAddHighlighting(disp, highlight, atompred)	highlights atoms with *OEHighlightBase*
OEAddHighlighting(disp, highlight, bondpred)	highlights bonds with *OEHighlightBase*
OEAddHighlighting(disp, highlight, atompred, bondpred)	highlights atoms and bonds with *OEHighlightBase*
OEAddHighlighting(disp, highlight, match)	highlights atoms and bonds of a match with *OEHighlightBase*
OEAddHighlighting(disp, highlight, abset)	highlights atoms and bonds of a set with *OEHighlightBase*
OEAddHighlighting(disp, highlight, subsearch)	highlights atoms and bonds of substructure matches *OEHighlightBase*

Table 46: **The overloaded versions of the OEAddHighlighting function using color and style**

Link	Description
OEAddHighlighting(disp, color, style, atompred)	highlights atoms with color and style
OEAddHighlighting(disp, color, style, bondpred)	highlights bonds with color and style
OEAddHighlighting(disp, color, style, atompred, bondpred)	highlights atoms and bonds with color and style
OEAddHighlighting(disp, color, style, match)	highlights atoms and bonds of a match with color and style
OEAddHighlighting(disp, color, style, abset)	highlights atoms and bonds of a set with color and style
OEAddHighlighting(disp, color, style, subsearch)	highlights atoms and bonds of substructure matches

Parameters :

disp
The *OE2DMolDisplay* object of which atoms add/or bonds being highlighted.

highlight
The *OEHighlightBase* object that specifies the style of the highlighting.

color
The OEColor object that specifies the color of the highlighting.

style
The style of the highlighting from the `OEHighlightStyle` namespace.

atompred
`OEAddHighlighting` function highlights only atoms for which the given atom predicate returns *true*.

bondpred
`OEAddHighlighting` function highlights only bonds for which the given bond predicate returns *true*.

match
: *OEAddHighlighting* function highlights the target atoms and bonds of the OEMatchBase object.

abset
: Stores the atoms and bonds being highlighted.

subsearch
: The substructure search object of the matches being highlighted.

Highlights atom and/or bonds with the style implemented in the given *OEHighlightBase* object

The atoms and bonds being highlighted can be specified either by a predicate or an OEMatchBase object or an OEAtomBondSet object.

The following code snippets show different ways to highlight any 6 membered aromatic ring in a molecule using the *OEHighlightByBallAndStick* highlighting class. They all generate the image shown in *Figure: Example of highlighting 6-membered aromatic rings*.

1. Highlighting using **atom** and **bond predicates**:

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OEHighlightBase& highlight,
                       const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪atompred)
```

Highlights atoms with the style implemented in the given *OEHighlightBase* object. The atoms being highlighted are specified by given atom predicate.

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OEHighlightBase& highlight,
                       const OESystem::OEUnaryPredicate<OEChem::OEBondBase>&␣
↪bondpred)
```

Highlights bonds with the style implemented in the given *OEHighlightBase* object. The bonds being highlighted are specified by given bond predicates.

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OEHighlightBase& highlight,
                       const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪atompred,
                       const OESystem::OEUnaryPredicate<OEChem::OEBondBase>&␣
↪bondpred)
```

Highlights atoms and bonds with the style implemented in the given *OEHighlightBase* object. The atoms and bonds being highlighted are specified by given atom and bond predicates, respectively.

Example: (See image generated in *Example of highlighting 6-membered aromatic rings*)

```
static class Pred6MemAromAtom extends OEUnaryAtomPred {
    @Override
    public boolean constCall(OEAtomBase atom) {
        return atom.IsAromatic() && oechem.OEAtomIsInAromaticRingSize(atom, 6);
    }

    @Override
    public OEUnaryAtomBoolFunc CreateCopy() {
        OEUnaryAtomBoolFunc copy = new Pred6MemAromAtom();
```

(continues on next page)

(continued from previous page)

```
        copy.swigReleaseOwnership();
        return copy;
    }
}

static class Pred6MemAromBond extends OEUnaryBondPred {
    @Override
    public boolean constCall(OEBondBase bond) {
        return bond.IsAromatic() && oechem.OEBondIsInAromaticRingSize(bond, 6);
    }

    @Override
    public OEUnaryBondBoolFunc CreateCopy() {
        OEUnaryBondBoolFunc copy = new Pred6MemAromBond();
        copy.swigReleaseOwnership();
        return copy;
    }
}

private static void OEAddHighlighting_Predicate(OE2DMolDisplay disp) {
    OEHighlightByBallAndStick highlightstyle = new OEHighlightByBallAndStick(oechem.
↪getOELightGreen());
    oedepict.OEAddHighlighting(disp, highlightstyle, new Pred6MemAromAtom());
    oedepict.OEAddHighlighting(disp, highlightstyle, new Pred6MemAromBond());
}

private static void OEAddHighlighting_AtomAndBondPredicate(OE2DMolDisplay disp) {
    OEHighlightByBallAndStick highlightstyle = new OEHighlightByBallAndStick(oechem.
↪getOELightGreen());
    oedepict.OEAddHighlighting(disp, highlightstyle, new Pred6MemAromAtom(), new␣
↪Pred6MemAromBond());
}
```

2. Highlighting using an OEMatchBase object that is initialized by substructure search or maximum common substructure search:

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OEHighlightBase& highlight,
                       const OEChem::OEMatchBase& match)
```

Example: (See image generated in *Example of highlighting 6-membered aromatic rings*)

```
private static void OEAddHighlighting_OEMatch(OE2DMolDisplay disp) {
    OESubSearch subs = new OESubSearch("a1aaaaa1");
    boolean unique = true;
    OEHighlightByBallAndStick highlightstyle = new OEHighlightByBallAndStick(oechem.
↪getOELightGreen());
    for (OEMatchBase match : subs.Match(disp.GetMolecule(), unique)) {
        oedepict.OEAddHighlighting(disp, highlightstyle, match);
    }
}
```

3. Highlighting using an OEAtomBondSet object that stores the atoms and bonds being highlighted:

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OEHighlightBase& highlight,
                       const OEChem::OEAtomBondSet& abset)
```

Example: (See image generated in *Example of highlighting 6-membered aromatic rings*)

```
private static void OEAddHighlighting_OEAtomBondSet(OE2DMolDisplay disp) {
    OEMolBase mol = disp.GetMolecule();
    OEAtomBondSet abset = new OEAtomBondSet(mol.GetAtoms(new Pred6MemAromAtom()),
            mol.GetBonds(new Pred6MemAromBond()));
    OEHighlightByBallAndStick highlightstyle = new OEHighlightByBallAndStick(oechem.
↪getOELightGreen());
    oedepict.OEAddHighlighting(disp, highlightstyle, abset);
}
```

4. Highlighting using an OESubSearch of which matches are being highlighted:

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OEHighlightBase& highlight,
                       const OEChem::OESubSearch& subsearch)
```

Example: (See image generated in *Example of highlighting 6-membered aromatic rings*)

```
private static void OEAddHighlighting_OESubSearch(OE2DMolDisplay disp) {
    OESubSearch subsearch = new OESubSearch("a1aaaaa1");
    OEHighlightByBallAndStick highlightstyle = new OEHighlightByBallAndStick(oechem.
↪getOELightGreen());
    oedepict.OEAddHighlighting(disp, highlightstyle, subsearch);
}
```

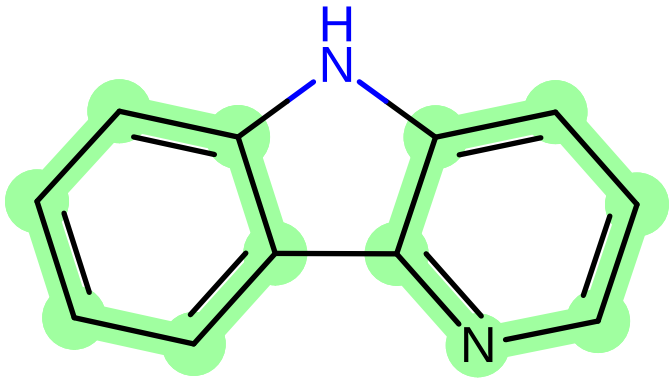


Fig. 336: **Example of highlighting 6-membered aromatic rings**

See also

- *OEHighlightByColor* class

- *OEHighlightByCogwheel* class
- *OEHighlightByBallAndStick* class
- *OEHighlightByStick* class
- *OEHighlightByLasso* class
- *OEDrawHighlighting* function
- *Highlighting* chapter

Highlights atoms and/or bonds with the given color and style

The atoms and bonds being highlighted can be specified either by a predicate or an OEMatchBase object or an OEAtomBondSet object.

The following code snippets show three different ways to highlight any 6 membered aromatic ring in a molecule using the built-in highlighting styles. They all generate the image shown in *Figure: Example of highlighting 6-membered aromatic rings*.

1. Highlighting using **atom** and **bond predicates**:

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OESystem::OEColor& color,
                       unsigned int style,
                       const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪atompred)
```

Highlights atoms with the given color and style. The atoms being highlighted are specified by given atom predicate.

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OESystem::OEColor& color,
                       unsigned int style,
                       const OESystem::OEUnaryPredicate<OEChem::OEBondBase>&␣
↪bondpred)
```

Highlights bonds with the given color and style. The bonds being highlighted are specified by given bonds predicate.

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OESystem::OEColor& color,
                       unsigned int style,
                       const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪atompred,
                       const OESystem::OEUnaryPredicate<OEChem::OEBondBase>&␣
↪bondpred)
```

Highlights atoms and bonds with the given color and style. The atoms and bonds being highlighted are specified by given atom and bond predicates, respectively.

Example: (See image generated in *Example of highlighting 6-membered aromatic rings*)

```
static class Pred6MemAromAtom extends OEUnaryAtomPred {
    @Override
    public boolean constCall(OEAtomBase atom) {
```

(continues on next page)

(continued from previous page)

```
        return atom.IsAromatic() && oechem.OEAtomIsInAromaticRingSize(atom, 6);
    }

    @Override
    public OEUnaryAtomBoolFunc CreateCopy() {
        OEUnaryAtomBoolFunc copy = new Pred6MemAromAtom();
        copy.swigReleaseOwnership();
        return copy;
    }
}

static class Pred6MemAromBond extends OEUnaryBondPred {
    @Override
    public boolean constCall(OEBondBase bond) {
        return bond.IsAromatic() && oechem.OEBondIsInAromaticRingSize(bond, 6);
    }

    @Override
    public OEUnaryBondBoolFunc CreateCopy() {
        OEUnaryBondBoolFunc copy = new Pred6MemAromBond();
        copy.swigReleaseOwnership();
        return copy;
    }
}

private static void OEAddHighlighting_Predicate(OE2DMolDisplay disp) {
    oedepict.OEAddHighlighting(disp, oechem.getOEDarkGreen(), OEHighlightStyle.
↪Color, new Pred6MemAromAtom());
    oedepict.OEAddHighlighting(disp, oechem.getOEDarkGreen(), OEHighlightStyle.
↪Color, new Pred6MemAromBond());
}

private static void OEAddHighlighting_AtomAndBondPredicate(OE2DMolDisplay disp) {
    oedepict.OEAddHighlighting(disp, oechem.getOEDarkGreen(), OEHighlightStyle.
↪Color,
    new Pred6MemAromAtom(), new Pred6MemAromBond());
}
```

2. Highlighting using an OEMatchBase object that is initialized by substructure search or maximum common substructure search:

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OESystem::OEColor& color,
                       unsigned int style,
                       const OEChem::OEMatchBase& match)
```

Example: (See image generated in *Example of highlighting 6-membered aromatic rings*)

```
private static void OEAddHighlighting_OEMatch(OE2DMolDisplay disp) {
    OESubSearch subs = new OESubSearch("a1aaaaa1");
    boolean unique = true;
    for (OEMatchBase match : subs.Match(disp.GetMolecule(), unique)) {
```

(continues on next page)

(continued from previous page)

```
        oedepict.OEAddHighlighting(disp, oechem.getOEDarkGreen(), OEHighlightStyle.
↪Color, match);
    }
}
```

3. Highlighting using an OEAtomBondSet object that stores the atoms and bonds being highlighted:

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OESystem::OEColor& color,
                       unsigned int style,
                       const OEChem::OEAtomBondSet& abset)
```

Example: (See image generated in *Example of highlighting 6-membered aromatic rings*)

```
private static void OEAddHighlighting_OEAtomBondSet(OE2DMolDisplay disp) {
    OEMolBase mol = disp.GetMolecule();
    OEAtomBondSet abset = new OEAtomBondSet(mol.GetAtoms(new Pred6MemAromAtom()),
            mol.GetBonds(new Pred6MemAromBond()));
    oedepict.OEAddHighlighting(disp, oechem.getOEDarkGreen(), OEHighlightStyle.
↪Color, abset);
}
```

4. Highlighting using an OESubSearch of which matches are being highlighted:

```
void OEAddHighlighting(OE2DMolDisplay& disp,
                       const OESystem::OEColor& color,
                       unsigned int style,
                       const OEChem::OESubSearch& subsearch)
```

Example: (See image generated in *Example of highlighting 6-membered aromatic rings*)

```
private static void OEAddHighlighting_OESubSearch(OE2DMolDisplay disp) {
    OESubSearch subsearch = new OESubSearch("a1aaaaa1");
    oedepict.OEAddHighlighting(disp, oechem.getOEDarkGreen(), OEHighlightStyle.
↪Color, subsearch);
}
```

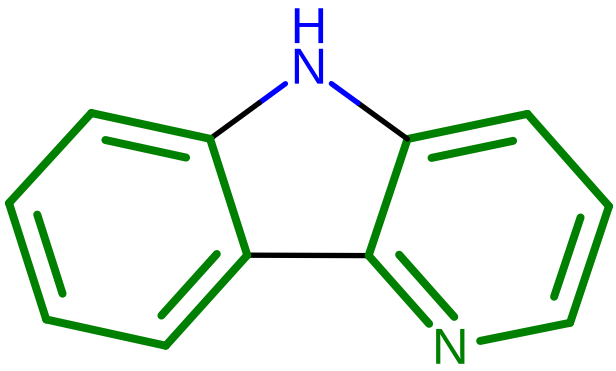


Fig. 337: **Example of highlighting 6-membered aromatic rings**

See also

- OEColor class
- *OEHighlightStyle* namespace
- *Highlighting* chapter

4.3.5 OEAddHighlightOverlay

```
void OEAddHighlightOverlay(OE2DMolDisplay& disp,
                           const OEHighlightOverlayBase& highlight,
                           OESystem::OEIter<OEChem::OEMatchBase>& matches)
void OEAddHighlightOverlay(OE2DMolDisplay& disp,
                           const OEHighlightOverlayBase& highlight,
                           OESystem::OEIter<OEChem::OEAtomBondSet>& absets)
void OEAddHighlightOverlay(OE2DMolDisplay& disp,
                           const OEHighlightOverlayBase& highlight,
                           OESystem::OEIterBase<OEChem::OEMatchBase>* matches)
void OEAddHighlightOverlay(OE2DMolDisplay& disp,
                           const OEHighlightOverlayBase& highlight,
                           OESystem::OEIterBase<OEChem::OEAtomBondSet>* absets)
void OEAddHighlightOverlay(OE2DMolDisplay& disp,
                           const OEHighlightOverlayBase& highlight,
                           const std::vector<OEChem::OEAtomBondSet>& absets)
```

Highlights a set of atoms and/or bonds of the displayed molecule.

Parameters :

disp
The *OE2DMolDisplay* object of which atoms add/or bonds being highlighted.

highlight
The *OEHighlightOverlayBase* object that specifies the style of the highlighting.

matches
The iterator of OEMatchBase objects of which target atoms and bonds being highlighted.

absets
The iterator of OEAtomBondSet objects of which atoms and bonds being highlighted.

Note

Each pattern will be highlighted with a different color returned by the *OEHighlightOverlayBase.GetColors* method.

See also

- *OEHighlightOverlayBase* class
- *OEHighlightOverlayStyle* namespace
- *Highlighting Overlapped Patterns* section

4.3.6 OEAddInteractiveIcon

```
void OEAddInteractiveIcon(OE2DMolDisplay& disp,
                          unsigned int loc = OEIconLocation::Default,
                          double scale = 1.0)
void OEAddInteractiveIcon(OEImageBase& image,
                          unsigned int loc = OEIconLocation::Default,
                          double scale = 1.0)
```

Adds an icon to an image or a molecule display that indicates that the image is interactive.

loc
This value has to be from the *OEIconLocation* namespace.

scale
The multiplier that can be used to increase or decrease the size of the icon. This value has to be in the range of `[0.5, 2.0]`.

4.3.7 OEAddLabel

Adds label to an image or molecule display.

Table 47: **The overloaded versions of the OEAddLabel function**

Link	Description
OEAddLabel(image, position, label)	adding label to an image
OEAddLabel(disp, label, match)	adding label to atoms of a match
OEAddLabel(disp, label, abset)	adding label to set of atoms
OEAddLabel(disp, label, atompred)	adding label to atoms specified by a predicate

Note

When adding label associated with atoms it is recommended to use the *OEAddLabel* along with the *OEAddHighlighting* function for visual clarity.

```
void OEAddLabel(OEImageBase& image, const OE2DPoint& center,
                const OEHighlightLabel& label)
```

image
The image in which the label is drawn.

center
The center of the label on the image.

label
The *OEHighlightLabel* object that stores properties that determine the styles of the label along with the text of the label itself.

Example:

```
private static void OEAddLabel_OEImage(OEImageBase image) {

    OEHighlightLabel label = new OEHighlightLabel("Hello!");
    oedepict.OEAddLabel(image, new OE2DPoint(50, 50), label);

    label.SetBoundingBoxPen(oedepict.getOETransparentPen());
    oedepict.OEAddLabel(image, new OE2DPoint(100, 50), label);

    label.SetBoundingBoxPen(oedepict.getOELightGreyPen());
    oedepict.OEAddLabel(image, new OE2DPoint(150, 50), label);
}
```

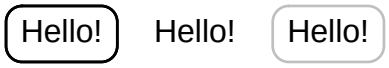


Fig. 338: **Example of adding labels to an image**

```
void OEAddLabel(OE2DMolDisplay& disp, const OEHighlightLabel& label,
                const OEChem::OEMatchBase& match)
```

Adds label to atoms of an OEMatchBase object that can be initialized by substructure search or maximum common substructure search.

disp
The *OE2DMolDisplay* object on which the label is going to be positioned.

label
The *OEHighlightLabel* object that stores properties that determine the styles of the label along with the text of the label itself.

match
The label is positioned based on the target atoms of the OEMatchBase object.

Example:

```
private static void OEAddLabel_OEMatch(OE2DMolDisplay disp) {
    OEMolBase mol = disp.GetMolecule();

    OESubSearch subs = new OESubSearch("a1aaaaa1");
    boolean unique = true;
    OEHighlightByBallAndStick highlight = new OEHighlightByBallAndStick(oechem.
↪getOELightGreen());
    for (OEMatchBase match : subs.Match(mol, unique)) {
        oedepict.OEAddHighlighting(disp, highlight, match);
        OEHighlightLabel label = new OEHighlightLabel("aromatic", oechem.
↪getOELightGreen());
        oedepict.OEAddLabel(disp, label, match);
    }
}
```

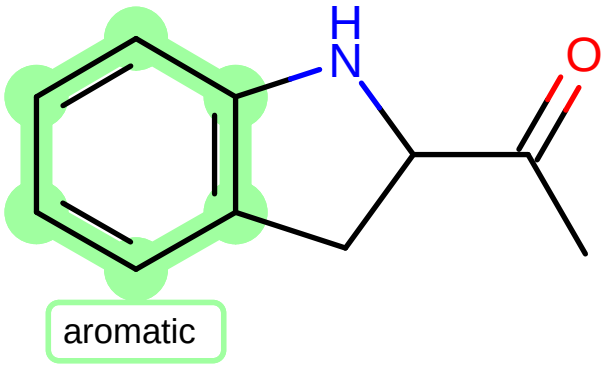


Fig. 339: **Example of adding label to 6-membered aromatic ring**

```
void OEAddLabel(OE2DMolDisplay& disp, const OEHighlightLabel& label,
                const OEChem::OEAtomBondSet& abset)
```

Adds label to atoms stored in the OEAtomBondSet object.

disp
The *OE2DMolDisplay* object on which the label is going to be positioned.

label
The *OEHighlightLabel* object that stores properties that determine the styles of the label along with the text of the label itself.

abset
The label is positioned based on the atoms stored in the OEAtomBondSet object.

Example:

```
private static void OEAddLabel_OEAtomBondSet(OE2DMolDisplay disp) {
    OEMolBase mol = disp.GetMolecule();

    OEAtomBondSet ringset = new OEAtomBondSet(mol.GetAtoms(new OEAtomIsInRing()),
          mol.GetBonds(new OEBondIsInRing()));
    OEHighlightByBallAndStick ringhighlight = new OEHighlightByBallAndStick(oechem.
↪getOELightGreen());
    oedepict.OEAddHighlighting(disp, ringhighlight, ringset);
    OEHighlightLabel ringlabel = new OEHighlightLabel("ring", oechem.getOELightGreen());
    oedepict.OEAddLabel(disp, ringlabel, ringset);

    OEAtomBondSet chainset = new OEAtomBondSet(mol.GetAtoms(new OEAtomIsInChain()),
                                               mol.GetBonds(new OEBondIsInChain()));
    OEHighlightByBallAndStick chainhighlight = new OEHighlightByBallAndStick(oechem.
↪getOEBlueTint());
    oedepict.OEAddHighlighting(disp, chainhighlight, chainset);
    OEHighlightLabel chainlabel = new OEHighlightLabel("chain", oechem.getOEBlueTint());
    oedepict.OEAddLabel(disp, chainlabel, chainset);
}
```

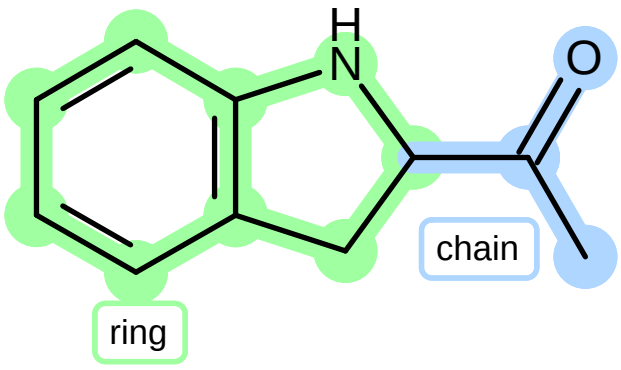


Fig. 340: **Example of adding label to ring and chain components of a molecule**

```
void OEAddLabel(OE2DMolDisplay& disp, const OEHighlightLabel& label,
                const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& atompred)
```

Adds label to atoms for which the given atom predicate returns *true*.

disp
: The *OE2DMolDisplay* object on which the label is going to be positioned.

label
: The *OEHighlightLabel* object that stores properties that determine the styles of the label along with the text of the label itself.

atompred
: The label is positioned based on the atoms for which the given atom predicate returns *true*.

Example:

```
private static void OEAddLabel_Predicate(OE2DMolDisplay disp) {
    OEHighlightByBallAndStick ringhighlight = new OEHighlightByBallAndStick(oechem.
↪getOELightGreen());
    oedepict.OEAddHighlighting(disp, ringhighlight, new OEAtomIsInRing(), new␣
↪OEBondIsInRing());
    OEHighlightLabel ringlabel = new OEHighlightLabel("ring", oechem.getOELightGreen());
    oedepict.OEAddLabel(disp, ringlabel, new OEAtomIsInRing());

    OEHighlightByBallAndStick chainhighlight = new OEHighlightByBallAndStick(oechem.
↪getOEBlueTint());
    oedepict.OEAddHighlighting(disp, chainhighlight, new OEAtomIsInChain(), new␣
↪OEBondIsInChain());
    OEHighlightLabel chainlabel = new OEHighlightLabel("chain", oechem.getOEBlueTint());
    oedepict.OEAddLabel(disp, chainlabel, new OEAtomIsInChain());
}
```

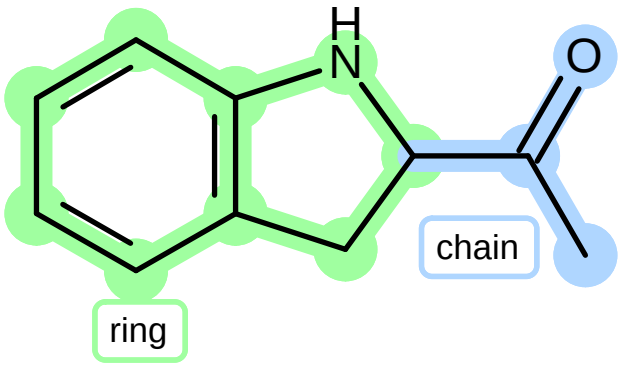


Fig. 341: **Example of adding label to ring and chain components of a molecule**

4.3.8 OEAddOpenEyeIcon

```
void OEAddOpenEyeIcon(OEImageBase& image,
                      unsigned int loc = OEIconLocation::Default,
                      double scale = 1.0)
void OEAddOpenEyeIcon(OE2DMolDisplay& disp,
                      unsigned int loc = OEIconLocation::Default,
                      double scale = 1.0)
```

Adds an OpenEye logo to an image or a molecule display.

loc
: This value has to be from the *OEIconLocation* namespace.

scale
: The multiplier that can be used to increase or decrease the size of the icon. This value has to be in the range of `[0.5, 2.0]`.

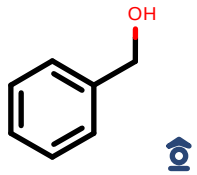


Fig. 342: **Example of using the OEAddOpenEyeIcon function**

4.3.9 OEAddSVGClickEvent

Table 48: **The overloaded versions of the OEAddSVGClickEvent function**

Link	Description
OEAddSVGClickEvent(disp, adisp, message)	adding click event to *OE2DAtomDisplay*
OEAddSVGClickEvent(disp, bdisp, message)	adding click event to *OE2DBondDisplay*
OEAddSVGClickEvent(svggroup, message)	adding click event to *OESVGGroup* (low-level function)

```
bool OEAddSVGClickEvent(OE2DMolDisplay& disp, OE2DAtomDisplay* adisp,
                        const std::string& message)
bool OEAddSVGClickEvent(OE2DMolDisplay& disp, OE2DBondDisplay* bdisp,
                        const std::string& message)
```

Adds a click event to an atom or bond in an `.svg` image. When clicking at the position defined by the given atom or bond display, an event is triggered with the given message.

Note

This functionality is **only available** for `.svg` image format.

See also

- *Generating Interactive SVG Images* chapter including the *How to Catch SVG Events* section

disp
The *OE2DMolDisplay* object that holds the data necessary to depict the molecule with which it is initialized.

adisp
The *OE2DAtomDisplay* object that defines the mouse click position.

bdisp
The *OE2DBondDisplay* object that defines the mouse click position

message
The text that will be emitted when an event is triggered on mouse click.

Example for adding event for atom displays

```
int width  = 400;
int height = 200;

OEImage image = new OEImage(width, height);

OEGraphMol mol = new OEGraphMol();
oechem.OESmilesToMol(mol, "Cc1cccnc1/C=C/[C@H](C(=O)O)O");
oedepict.OEPrepareDepiction(mol);

OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, OEScale.AutoScale);
OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);

for (OE2DAtomDisplay adisp : disp.GetAtomDisplays()) {
    OEAtomBase atom = adisp.GetAtom();
    String message = "atom idx=" + Integer.toString(atom.GetIdx());
    oedepict.OEAddSVGClickEvent(disp, adisp, message);
}
oedepict.OERenderMolecule("AddAtomClickEvent.svg", disp);
```

Example for adding event for bond displays

```
int width  = 400;
int height = 200;

OEImage image = new OEImage(width, height);

OEGraphMol mol = new OEGraphMol();
oechem.OESmilesToMol(mol, "Cc1cccnc1/C=C/[C@H](C(=O)O)O");
oedepict.OEPrepareDepiction(mol);

OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, OEScale.AutoScale);
OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);

for (OE2DBondDisplay bdisp : disp.GetBondDisplays()) {
    OEBondBase bond = bdisp.GetBond();
    String message = "bond idx=" + Integer.toString(bond.GetIdx());
    oedepict.OEAddSVGClickEvent(disp, bdisp, message);
}
oedepict.OERenderMolecule("AddBondClickEvent.svg", disp);
```

See also

- *OE2DMolDisplay* class
- *OE2DAtomDisplay* and *OE2DBondDisplay* classes

```
bool OEAddSVGClickEvent(OESVGGroup* svggroup,
                        const std::string& message)
```

Adds a click event to a specific SVG group in `.svg` image. When clicking at the elements drawn inside the given *OESVGGroup* object, an event is triggered with the given message

Note

This functionality is **only available** for `.svg` image format.

Hint

The *OEAddSVGClickEvent* function should always be called prior to pushing / popping the *OESVGGroup* object.

Example:

The following example creates an `.svg` image with a red rectangle and a blue circle. Both drawing elements are rendered inside *OESVGGroup* objects that are associated with messages `clicked on rectangle` and `clicked on circle`, respectively. When the generated `.svg` image is displayed in a HTTP server, clicking on the rectangle and the circle triggers events with the corresponding messages.

```
OESVGGroup group_rectangle = image.NewSVGGroup("rectangle");

oedepict.OEAddSVGClickEvent(group_rectangle, "clicked on rectangle");

image.PushGroup(group_rectangle);
image.DrawRectangle(new OE2DPoint(30, 30), new OE2DPoint(70, 70), oedepict.
↪getOERedBoxPen());
image.PopGroup(group_rectangle);

OESVGGroup group_circle = image.NewSVGGroup("circle");

oedepict.OEAddSVGClickEvent(group_circle, "clicked on circle");

image.PushGroup(group_circle);
image.DrawCircle(new OE2DPoint(150, 50), 30, oedepict.getOEBlueBoxPen());
image.PopGroup(group_circle);

oedepict.OEWriteImage("AddSVGClickEvent.svg", image);
```

See also

- *Generating Interactive SVG Images* chapter including the *How to Catch SVG Events* section that shows how to test the `.svg` image file generated by this example.
- *OEAddSVGHover* function
- *OEAddSVGToggle* function

4.3.10 OEAddSVGHover

```
bool OEAddSVGHover(OESVGGroup* area, OESVGGroup* target)
```

Adds a hover effect to the objects drawn between the two given groups. While hovering the mouse over objects drawn inside the **area** group the objects drawn in the **target** group are displayed.

Note

This functionality is **only available** for `.svg` image format.

See also

- *Generating Interactive SVG Images* chapter
- *OESVGGroup* class
- *OEAddSVGClickEvent* function
- *OEAddSVGToggle* function

Example:

```
OESVGGroup hover_area = image.NewSVGGroup("hover_area");
OESVGGroup target_area = image.NewSVGGroup("hover_target_area");

oedepict.OEAddSVGHover(hover_area, target_area);

image.PushGroup(hover_area);
image.DrawRectangle(new OE2DPoint(30, 30), new OE2DPoint(70, 70), oedepict.
↪getOERedBoxPen());
image.PopGroup(hover_area);

image.PushGroup(target_area);
image.DrawCircle(new OE2DPoint(150, 50), 30, oedepict.getOEBlueBoxPen());
image.PopGroup(target_area);

oedepict.OEAddInteractiveIcon(image);

oedepict.OEWriteImage("AddSVGHover.svg", image);
```

Hint

The *OEAddSVGHover* function should always be called prior to pushing / popping the *OESVGGroup* objects.

4.3.11 OEAddSVGToggle

```
bool OEAddSVGToggle(OESVGGroup* area, OESVGGroup* target,
                    bool on = false)
```

Adds a toggle effect to the objects drawn between the two given groups. When clicking on the objects drawn inside the **area** group the objects drawn in the **target** group appear / disappear.

on
: The parameter that determine whether the objects drawn inside the **target** group are displayed initially.

Note

This functionality is **only available** for `.svg` image format.

See also

- *Generating Interactive SVG Images* chapter
- *OESVGGroup* class
- *OEAddSVGClickEvent* function
- *OEAddSVGHover* function

Example:

```
OESVGGroup toggle_area = image.NewSVGGroup("toggle_area");
OESVGGroup target_area = image.NewSVGGroup("toggle_target_area");

oedepict.OEAddSVGToggle(toggle_area, target_area);

image.PushGroup(toggle_area);
image.DrawRectangle(new OE2DPoint(30, 30), new OE2DPoint(70, 70), oedepict.
↪getOERedBoxPen());
image.PopGroup(toggle_area);

image.PushGroup(target_area);
image.DrawCircle(new OE2DPoint(150, 50), 30, oedepict.getOEBlueBoxPen());
image.PopGroup(target_area);

oedepict.OEAddInteractiveIcon(image);

oedepict.OEWriteImage("AddSVGToggle.svg", image);
```

Hint

The *OEAddSVGToggle* function should always be called prior to pushing / popping the *OESVGGroup* objects.

4.3.12 OEAddWatermark

```
void OEAddWatermark(OEImageBase& image, const std::string& text,
                    double scale = 1.0)
void OEAddWatermark(OE2DMolDisplay& disp, const std::string& text,
                    double scale = 1.0)
```

Adds a watermark to an image or a molecule display.

text
: The text of the watermark.

scale
: The multiplier that can be used to increase or decrease the font size of the watermark text. This value has to be

in the range of `[0.5, 2.0]`.

4.3.13 OEAnnotateDepictionProblems

```
bool OEAnnotateDepictionProblems(OE2DMolDisplay& disp)
```

Annotates various depiction problems on the molecule. Currently it can mark double bonds which are not depicted correctly according to their cis/trans stereo specifications.

disp
The *OE2DMolDisplay* object being annotated.

The following code snippet show how to use the *OEAnnotateDepictionProblems* function. See the generated image in *Figure: Example of incorrect trans double bond depiction*. The highlighting indicates that the double bond specified in the input SMILES string as trans is depicted as cis.

```
OEGraphMol mol = new OEGraphMol();
oechem.OESmilesToMol(mol, "C1CCC2C/C=C/CCCC2C1");
oedepict.OEPrepareDepiction(mol);

int width  = 200;
int height = 200;
double scale  = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);

OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
oedepict.OEAnnotateDepictionProblems(disp);

oedepict.OERenderMolecule("OEAnnotateDepictionProblems.png", disp);
```

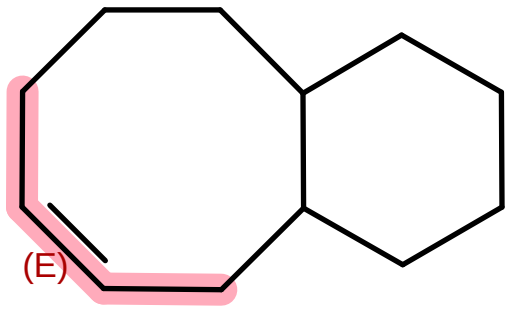


Fig. 343: **Example of incorrect trans double bond depiction**

Note

OEDepict TK uses built-in ring system templates when generating 2D coordinates. Generating 2D coordinates

which are inconsistent with cis/trans stereo bond specification can occur due to a missing ring template. The *OEPrepareDepiction* function throws a warning when this problem occurs.

4.3.14 OEAnnotateValenceProblems

```
bool OEAnnotateValenceProblems(OE2DMolDisplay& disp)
```

Annotates potential valence problems on the molecule. An atom is considered to have a valence problem when its bond count exceeds the maximum valence defined by the MDL valence model for its atomic number and formal charge.

Note

Aromaticity must be perceived (e.g.via `OEAssignAromaticFlags`) before calling this function. Without aromaticity perception, aromatic ring atoms may be incorrectly flagged as having valence problems because their bonds are treated as alternating single/double bonds.

disp
The *OE2DMolDisplay* object being annotated.

The following code snippet show how to use the *OEAnnotateValenceProblems* function. See the generated image in *Figure: Example of molecule with incorrect valences*.

```
OEGraphMol mol = new OEGraphMol();
oechem.OESmilesToMol(mol, "C[N+](=C)(C)C1CCCCl1");
oedepict.OEPrepareDepiction(mol);

int width  = 200;
int height = 200;
double scale  = OEScale.AutoScale;
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, scale);

OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
oedepict.OEAnnotateValenceProblems(disp);

oedepict.OERenderMolecule("OEAnnotateValenceProblems.png", disp);
```

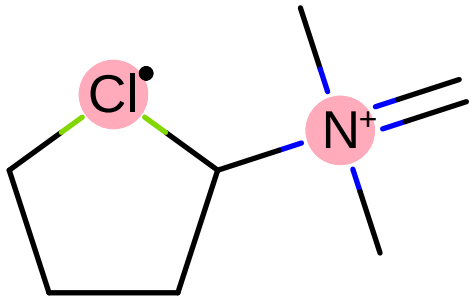


Fig. 344: **Example of molecule with incorrect valences**

Note

The *OEAnnotateValenceProblems* function uses the MDL Valence Model.

4.3.15 OEConfigure2DMolDisplayOptions

```
bool OEConfigure2DMolDisplayOptions(OESystem::OEInterface& itf,
                                    unsigned int config = OE2DMolDisplaySetup::All)
```

Configures the given interface to add molecule depiction style parameters.

itf
The interface being configured.

config
The option that specifies which parameters will be added to the interface This value has to be from the *OE2DMolDisplaySetup* namespace.

See also

- OEInterface class in the **OEChem TK** manual
- *OE2DMolDisplaySetup* namespace
- *OESetup2DMolDisplayOptions* function

OEConfigure2DMolDisplayOptions is the utility function that allows to easily generate consistent command line interface for depiction examples and applications. For each command line options the generated interface will include information about related APIs in **OEDepict TK**, For example:

```
prompt> java Mol2Img --help -aromstyle
```

will reveals that the *-aromstyle* command line parameter is related to the *OE2DMolDisplayOptions.SetAromaticStyle* method and the *OEAromaticStyle* namespace (see the "Details" section below):

```
Contents of parameter -aromstyle
 Aliases : -astl
 Type : string
 Allow list : false
 Default : Kekule
 Simple : true
 Required : false
 Legal values : Kekule Circle Dash
 Brief : Aromatic ring display style
 Detail

API : OE2DMolDisplayOptions::SetAromaticStyle method and OEAromaticStyle namespace
```

When generating an html help file:

```
prompt> java Mol2Img --help html
Creating 'mol2img.py_help.html'
```

The generated 'mol2img.py_help.html' file will contain hyperlinks to related APIs in the **OEDepict TK** documentation.

See also

- *Depicting a Single Molecule*

4.3.16 OEConfigureColor

```
bool OEConfigureColor(OESystem::OEInterface& itf,
                      const std::string& name,
                      const std::string& alias,
                      const std::string& brief,
                      std::string defcolor = "red")
```

Adds a *color* parameter to the given interface.

itf
: The interface being configured.

name
: The name of the color parameter.

alias
: The alternative *i.e.* alias name of the parameter.

brief
: The brief description of the parameter.

defcolor
: The default color of the parameter.

Example:

```
OEInterface itf = new OEInterface();
oedepict.OEConfigureColor(itf, "mycolor", "c", "Example of using OEConfigureColor", "blue
↪");
```

The above snippet will generate the following parameter:

```
Contents of parameter mycolor
    Aliases : c
    Type : string
    Allow list : false
    Default : blue
    Simple : true
    Required : false
    Legal values : black blue bluetint brass brown copper cyan darkblue
       darkbrown darkcyan darkgreen darkgrey darkmagenta darkorange
       darkpurple darkred darkrose darksalmon darkyellow gold green
       greenblue greentint grey hotpink lightblue lightbrown lightgreen
       lightgrey lightorange lightpurple lightsalmon limegreen magenta
       mediumblue mediumbrown mediumgreen mediumorange mediumpurple
       mediumsalmon mediumyellow olivebrown olivegreen olivegrey pewter
       orange pink pinktint purple red redorange royalblue seagreen silver
       skyblue violet white yellow yellowtint
    Brief : Example of using OEConfigureColor
```

See also

- OEInterface class in the **OEChem TK** manual
- *OEGetColor* function
- OEColor class

4.3.17 OEConfigureHighlightColor

```
bool OEConfigureHighlightColor(OESystem::OEInterface& itf,
                               std::string name = OEHighlightParamName::Color,
                               std::string alias = "-hcolor")
```

Adds a *highlighting color* parameter to the given interface.

itf
The interface being configured.

name
The name of the color parameter.

alias
The alternative *i.e.* alias name of the parameter.

See also

- OEInterface class in the **OEChem TK** manual
- OEColor class

- *OEHighlightSetup.Color* constant
- *OEHighlightParamName.Color* constant
- *OEGetHighlightColor* function

4.3.18 OEConfigureHighlightParams

```
bool OEConfigureHighlightParams(OESystem::OEInterface& itf,
                                unsigned int config = OEHighlightSetup::All)
```

Configures the given interface to add highlight style parameters.

itf
The interface being configured.

config
The option that specifies which parameters will be added to the interface. This value has to be from the *OEHighlightSetup* namespace.

See also

- OEInterface class in the **OEChem TK** manual
- *OEHighlightSetup* namespace
- *OEGetHighlightColor* function
- *OEGetHighlightStyle* function

4.3.19 OEConfigureHighlightStyle

```
bool OEConfigureHighlightStyle(OESystem::OEInterface& itf,
                               std::string name = OEHighlightParamName::Style,
                               std::string alias = "-hstyle")
```

Adds a *highlighting style* parameter to the given interface.

itf
The interface being configured.

name
The name of the style parameter.

alias
The alternative *i.e.* alias name of the parameter.

See also

- OEInterface class in the **OEChem TK** manual
- *OEHighlightSetup.Style* constant
- *OEHighlightParamName.Style* constant
- *OEGetHighlightStyle* function

- *OEHighlightStyle* namespace

4.3.20 OEConfigureImageGridNumColumns

```
bool OEConfigureImageGridNumColumns(OESystem::OEInterface& itf,
                                    std::string name = OEImageGridParamName::NumColumns,
                                    std::string alias = "-cols")
bool OEConfigureImageGridNumColumns(OESystem::OEInterface& itf,
                                    unsigned int defnumcols,
                                    std::string = OEImageGridParamName::NumColumns,
                                    std::string alias = "-cols")
```

Adds a parameter to the given interface to allows the configuration of the number of grid columns.

itf
The interface being configured.

defnumcols
The default number of columns.

name
The name of the parameter.

alias
The alternative *i.e.* alias name of the parameter.

See also

- OEInterface class in the **OEChem TK** manual
- *OEImageGridSetup.NumColumns* constant
- *OEImageGridParamName.NumColumns* constant
- *OEConfigureImageGridParams* function
- *OEGetImageGridNumColumns* function

4.3.21 OEConfigureImageGridNumRows

```
bool OEConfigureImageGridNumRows(OESystem::OEInterface& itf,
                                 std::string name = OEImageGridParamName::NumRows,
                                 std::string alias = "-rows")
bool OEConfigureImageGridNumRows(OESystem::OEInterface& itf,
                                 unsigned int defnumrows,
                                 std::string = OEImageGridParamName::NumRows,
                                 std::string alias = "-rows")
```

Adds a parameter to the given interface to allows the configuration of the number of grid rows.

itf
The interface being configured.

defnumrows
The default number of rows.

name
The name of the parameter.

alias
The alternative *i.e.* alias name of the parameter.

See also

- OEInterface class in the **OEChem TK** manual
- *OEImageGridSetup.NumRows* constant
- *OEImageGridParamName.NumRows* constant
- *OEConfigureImageGridParams* function
- *OEGetImageGridNumRows* function

4.3.22 OEConfigureImageGridParams

```
bool OEConfigureImageGridParams(OESystem::OEInterface& itf,
                                unsigned int config = OEImageGridSetup::All)
```

Configures the given interface to add highlight style parameters.

itf
The interface being configured.

config
The option that specifies which parameters will be added to the interface. This value has to be from the *OEImageSetup* namespace.

See also

- OEInterface class in the **OEChem TK** manual
- *OEImageSetup* namespace
- *OEConfigureImageGridNumColumns* function
- *OEConfigureImageGridNumRows* function

4.3.23 OEConfigureImageHeight

```
bool OEConfigureImageHeight(OESystem::OEInterface& itf,
                            double defheight = 0.0,
                            std::string = OEImageParamName::Height,
                            std::string alias = "-h")
```

Adds a parameter to the given interface to allows the configuration of the image height.

itf
The interface being configured.

defheight
Default height value.

name
The name of the parameter.

alias
The alternative *i.e.* alias name of the parameter.

See also

- OEInterface class in the **OEChem TK** manual
- *OEImageSetup.Height* constant
- *OEImageParamName.Height* constant
- *OEConfigureImageOptions* function
- *OEGetImageHeight* function

4.3.24 OEConfigureImageOptions

```
bool OEConfigureImageOptions(OESystem::OEInterface& itf,
                             unsigned int config = OEImageSetup::All)
```

Configures the given interface to add image parameters.

itf
The interface being configured.

config
The option that specifies which parameters will be added to the interface. This value has to be from the *OEImageSetup* namespace.

See also

- OEInterface class in the **OEChem TK** manual
- *OEImageSetup* namespace

4.3.25 OEConfigureImageWidth

```
bool OEConfigureImageWidth (OESystem::OEInterface& itf,
                            double defwidth = 0.0,
                            std::string = OEImageParamName::Width,
                            std::string alias = "-w")
```

Adds a parameter to the given interface to allows the configuration of the image width.

itf
The interface being configured.

defwidth
Default width value.

name
The name of the parameter.

alias
The alternative *i.e.* alias name of the parameter.

See also

- OEInterface class in the **OEChem TK** manual
- *OEImageSetup.Width* constant
- *OEImageParamName.Width* constant
- *OEConfigureImageOptions* function
- *OEGetImageWidth* function

4.3.26 OEConfigureMultiPageOrientation

```
bool OEConfigureMultiPageOrientation(OESystem::OEInterface& itf,
                                     std::string name =␣
↪OEMultiPageParamName::PageOrientation,
                                     std::string alias = "-pori")
```

Adds a parameter to the given interface to allows the configuration of the page orientation.

itf
The interface being configured.

name
The name of the parameter.

alias
The alternative *i.e.* alias name of the parameter.

See also

- OEInterface class in the **OEChem TK** manual
- *OEMultiPageSetup.PageOrientation* constant
- *OEMultiPageParamName.PageOrientation* constant
- *OEConfigureMultiPageParams* function
- *OEGetMultiPageOrientation* function

4.3.27 OEConfigureMultiPageParams

```
bool OEConfigureMultiPageParams(OESystem::OEInterface& itf,
                                unsigned int config = OEMultiPageSetup::All)
```

Configures the given interface to add multi-page parameters.

itf
The interface being configured.

config
The option that specifies which parameters will be added to the interface. This value has to be from the

OEMultiPageSetup namespace.

See also

- OEInterface class in the **OEChem TK** manual
- *OEMultiPageSetup* namespace
- *OEConfigureMultiPageOrientation* function
- *OEConfigureMultiPageSize* function

4.3.28 OEConfigureMultiPageSize

```
bool OEConfigureMultiPageSize(OESystem::OEInterface& itf,
                              std::string name = OEMultiPageParamName::PageSize,
                              std::string alias = "-psize")
```

Adds a parameter to the given interface to allows the configuration of the page size.

itf
The interface being configured.

name
The name of the parameter.

alias
The alternative *i.e.* alias name of the parameter.

See also

- OEInterface class in the **OEChem TK** manual
- *OEMultiPageSetup.PageSize* constant
- *OEMultiPageParamName.PageSize* constant
- *OEConfigureMultiPageParams* function
- *OEGetMultiPageSize* function

4.3.29 OEConfigurePrepareDepictionOptions

```
bool OEConfigurePrepareDepictionOptions(OESystem::OEInterface& itf,
                                        unsigned int config =␣
↪OEPrepareDepictionSetup::All)
```

Configures the given interface to add parameters for preparing molecules for depiction.

itf
The interface being configured.

config
The option that specifies which parameters will be added to the interface This value has to be from the *OEPrepareDepictionSetup* namespace.

See also

- OEInterface class in the **OEChem TK** manual
- *OEPrepareDepictionSetup* namespace
- *OESetupPrepareDepictionOptions* function

4.3.30 OEConfigureReportOptions

```
bool OEConfigureReportOptions(OESystem::OEInterface& itf,
                              unsigned int config = OEReportSetup::All)
```

Configures the given interface to add parameters for the report layout.

itf
The interface being configured.

config
The option that specifies which parameters will be added to the interface This value has to be from the *OEReportSetup* namespace.

See also

- OEInterface class in the **OEChem TK** manual
- *OEReportSetup* namespace
- *OESetupReportOptions* function

4.3.31 OECreateWin32GraphicsImage

Note

This function is only available on the Windows operating system through C++.

4.3.32 OEDepictCoordinates

```
bool OEDepictCoordinates(OEChem::OEMolBase& mol)
```

Assigns 2D coordinates to the given OEMolBase object by invoking the `OEGenerate2DCoordinates` function. The coordinates of each explicit atom are assigned using the `OEMolBase.SetCoords` method. This function automatically sets the dimensionality of the molecule to two, using the `OEMolBase.SetDimension` method.

> **Warning**
>
> `OEDepictCoordinates` is considered deprecated and will be removed in OEDepict 3.0. Please switch to using `OEGenerate2DCoordinates`.

4.3.33 OEDepictGetArch

```
const char* OEDepictGetArch()
```

4.3.34 OEDepictGetLicensee

```
bool OEDepictGetLicensee(std::string& licensee)
```

4.3.35 OEDepictGetPlatform

```
const char* OEDepictGetPlatform()
```

4.3.36 OEDepictGetRelease

```
const char* OEDepictGetRelease()
```

4.3.37 OEDepictGetSite

```
bool OEDepictGetSite(std::string& site)
```

4.3.38 OEDepictGetVersion

```
unsigned int OEDepictGetVersion()
```

4.3.39 OEDepictIsLicensed

```
bool OEDepictIsLicensed(const char* feature = 0,
                        unsigned int* expdate = 0)
```

Determines whether a valid license file is present. This function may be called without a legitimate run-time license to determine whether it is safe to call any **OEDepict TK** functionality.

The 'feature' argument can be used to check for a valid license to **OEDepict TK** along with the following features: 'python', 'java', or 'clr' (for CSharp).

The 'expdate' argument can be used to obtain the expiration date of the current **OEDepict TK** license.

> **See also**
>
> - `OEChemIsLicensed` function

4.3.40 OEDrawBorder

```
void OEDrawBorder(OEImageBase& image, const OEPen& pen)
```

Draws a border around the image with the given pen.

image
The image around which the border is drawn.

pen
The graphical properties of the border.

See code snippet below and the corresponding image in *Figure: Example of using the OEDrawBorder function*.

```
int width  = 100;
int height = 50;
OEImage image = new OEImage(width, height);

OEPen pen = new OEPen(oechem.getOEWhite(), oechem.getOELightGreen(), OEFill.Off, 4.0);
oedepict.OEDrawBorder(image, pen);

oedepict.OEWriteImage("OEDrawBorder.png", image);
```


Fig. 345: **Example of using the OEDrawBorder function**

See also

- *OEImageBase* class
- *OEPen* class
- *OEDrawCurvedBorder* function

4.3.41 OEDrawChevronArrow

```
void OEDrawChevronArrow(OEImageBase& image, const OE2DPoint& bgn,
                        const OE2DPoint& end, const OEPen& pen,
                        double arrowWidth = 5.0, double gapScale = 2.0)
```

Draws a chevron arrow. See example in *Figure: Example of drawing a chevron arrows*.

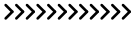


Fig. 346: **Example of drawing a chevron arrow**

image
The image in which the arrow is drawn.

bgn*, *end
The begin and end points of the arrow.

pen
The graphical properties of the arrow.

width
The width of the arrow.

gapScale
The multiplier that defines the space between the lines of the chevron.

See also

- *OEImageBase* class
- *OE2DPoint* class
- *OEPen* class

4.3.42 OEDrawCurvedArrow

```
void OEDrawCurvedArrow(OEImageBase& image, const OE2DPoint& center,
                       double bgnAngle, double endAngle,
                       double radius, bool clockwise, const OEPen& pen)
```

Draws a curved arrow. See example in *Figure: Example of drawing curved arrows*.

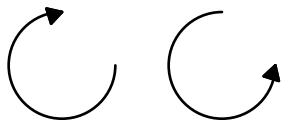


Fig. 347: **Example of drawing curved arrows**

image
The image in which the arrow is drawn.

center
The center of the arc.

bgnAngle*, *endAngle
The two endpoints of the arc (in degrees). Both angles are in degrees and their values have to be in a range of $[0.0°, 360.0°]$. Angles are interpreted such that $0.0°$ and $360.0°$ degrees are at the *12 o'clock* position, $90.0°$ degrees corresponds to *3 o'clock*, etc.

radius
The radius of the arc.

clockwise
The direction of the arrow.

pen
The graphical properties of the curved arrow.

See also

- *OEImageBase* class
- *OE2DPoint* class
- *OEPen* class

4.3.43 OEDrawCurvedBorder

```
void OEDrawCurvedBorder(OEImageBase& image, const OEPen& pen,
                        double radius)
```

Draws a round rectangle around the image with the given pen.

image
The image around which the border is drawn.

pen
The graphical properties of the border.

radius
The parameter that determines the roundedness of the corners of the rectangle.

See code snippet below and the corresponding image in *Figure: Example of using the OEDrawCurvedBorder function*.

```
int width  = 100;
int height = 50;
OEImage image = new OEImage(width, height);

OEPen pen = new OEPen(oechem.getOEWhite(), oechem.getOELightGreen(), OEFill.Off, 4.0);
oedepict.OEDrawCurvedBorder(image, pen, 15);

oedepict.OEWriteImage("OEDrawCurvedBorder.png", image);
```


Fig. 348: **Example of using the OEDrawCurvedBorder function**

See also

- *OEImageBase* class
- *OEPen* class
- *OEDrawBorder* function
- *OEDrawCurvedRectangle* function

4.3.44 OEDrawCurvedRectangle

```
void OEDrawCurvedRectangle(OEImageBase& image, const OE2DPoint& tl,
                           const OE2DPoint& br, const OEPen& pen,
                           double radius)
```

Draws a round rectangle with the given pen.

image
The image around which the border is drawn.

tl, br
The position of the top-left and bottom-right corner of the rectangle.

pen
The graphical properties of the border.

radius
The parameter that determines the roundedness of the corners of the rectangle.

See also

- *OEImageBase* class
- *OEPen* class
- *OEDrawBorder* function
- *OEDrawCurvedBorder* function

4.3.45 OEDrawEquilibriumArrow

```
void OEDrawEquilibriumArrow(OEImageBase& image, const OE2DPoint& bgn,
                            const OE2DPoint& end, const OEPen& pen)
```

Draws an equilibrium arrow between two points using the given pen. See example in *Figure: Example of drawing an equilibrium arrow*.

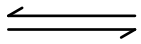


Fig. 349: **Example of drawing an equilibrium arrow**

image
The image in which the arrow is drawn.

bgn, end
The begin and end points of the arrow.

pen
The graphical properties of the arrow.

See also

- *OEImageBase* class

- *OE2DPoint* class
- *OEPen* class

4.3.46 OEDrawHighlighting

```
void OEDrawHighlighting(OEImageBase& image,
                        const OE2DMolDisplay& disp,
                        const OEHighlightBase& highlight,
                        const OEChem::OEAtomBondSet& abset)
```

Draws atom and bond highlighting into the image.

image
The image in which the highlighting is drawn.

disp
The *OE2DMolDisplay* object of which atoms add/or bonds being highlighted.

highlight
The *OEHighlightBase* object that specifies the style of the highlighting.

abset
Stores the atoms and bonds being highlighted.

Note

This function renders the highlighting into an image rather than into the molecule display as does the *OEAddHighlighting* function. Since this function does not change the molecule display it is **not available** for the *OEHighlightByColor* highlighting style.

See also

- *OEAddHighlighting* function

4.3.47 OEDrawLegendLayout

```
bool OEDrawLegendLayout(OELegendLayout& layout)
```

Draws the legend layout.

Note

Interactive legend can only be used in `.svg` image format.

See also

- *OELegendLayout* class
- *OELegendLayoutOptions* class

- *OELegendLayoutStyle* namespace
- *OELegendColorStyle* namespace
- *OELegendInteractiveStyle* namespace

Example

```
OELegendLayoutOptions opts = new OELegendLayoutOptions(OELegendLayoutStyle.
↪VerticalTopRight,
                                                       OELegendColorStyle.LightGreen,
                                                       OELegendInteractiveStyle.Toggle);
opts.SetButtonWidthScale(1.5);
opts.SetButtonHeightScale(1.5);
OELegendLayout legend = new OELegendLayout(image, "Legend", opts);

OEImage legend_area = legend.GetLegendArea();
oedepict.OEAddWatermark(legend_area, "This is the legend area");
oedepict.OEAddInteractiveIcon(image, OEIconLocation.BottomRight, 0.75);
oedepict.OEDrawLegendLayout(legend);

oedepict.OEWriteImage("LegendLayout.svg", image);
```

Hint

The *OEDrawLegendLayout* function should be called after rendering all other information to the image.

4.3.48 OEDrawReactionArrow

```
void OEDrawReactionArrow(OEImageBase& image, const OE2DPoint& bgn,
                         const OE2DPoint& end, const OEPen& pen)
```

Draws a reaction arrow between two points using the given pen. See example in *Figure: Example of drawing a reaction arrow*.



Fig. 350: **Example of drawing a reaction arrow**

image
: The image in which the arrow is drawn.

bgn*, *end
: The begin and end points of the arrow.

pen
: The graphical properties of the arrow.

See also

- *OEImageBase* class
- *OE2DPoint* class
- *OEPen* class

4.3.49 OEDrawResonanceArrow

```
void OEDrawResonanceArrow(OEImageBase& image, const OE2DPoint& bgn,
                          const OE2DPoint& end, const OEPen& pen)
```

Draws a resonance arrow between two points using the given pen. See example in *Figure: Example of drawing a resonance arrow*.



Fig. 351: **Example of drawing a resonance arrow**

image
The image in which the arrow is drawn.

bgn*, *end
The begin and end points of the arrow.

pen
The graphical properties of the arrow.

See also

- *OEImageBase* class
- *OE2DPoint* class
- *OEPen* class

4.3.50 OEDrawSVGHoverText

```
bool OEDrawSVGHoverText(OE2DMolDisplay& disp, OE2DAtomDisplay* adisp,
                        const std::string& text, const OEFont& font)
bool OEDrawSVGHoverText(OE2DMolDisplay& disp, OE2DBondDisplay* bdisp,
                        const std::string& text, const OEFont& font)
```

Creates and interactive effect in `.svg` images. The given text will appear when the mouse is hovering over the position defined by the given atom / bond display.

Note

This functionality is **only available** for `.svg` image format. In other image formats it has no effect.

The generated `svg` image should be included into and HTML page with the SVG MIME type.

```
<object data="<imagename>.svg" type="image/svg+xml"></object>
```

See also

- *Generating Interactive SVG Images* chapter

disp
The *OE2DMolDisplay* object that holds the data necessary to depict the molecule with which it is initialized.

adisp
The *OE2DAtomDisplay* object that defines where the text will appear.

bdisp
The *OE2DBondDisplay* object that defines where the text will appear.

text
The text that appears when the cursor is hovered over the given position.

font
The graphical properties of the font used to draw the text.

```
int width  = 400;
int height = 200;

OEImage image = new OEImage(width, height);

OEGraphMol mol = new OEGraphMol();
oechem.OESmilesToMol(mol, "Cc1cccnc1/C=C/[C@H](C(=O)O)O");
oedepict.OEPrepareDepiction(mol);

OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, OEScale.AutoScale);
opts.SetMargins(10);
OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);

OEFont font = new OEFont(OEFontFamily.Default, OEFontStyle.Default, 12,
                         OEAlignment.Center, oechem.getOEDarkRed());

for (OE2DAtomDisplay adisp : disp.GetAtomDisplays()) {
    OEAtomBase atom = adisp.GetAtom();
    String hovertext = "atom idx=" + Integer.toString(atom.GetIdx());
    oedepict.OEDrawSVGHoverText(disp, adisp, hovertext, font);
}
oedepict.OERenderMolecule("HoverAtomText.svg", disp);
```

There is no image available for the PDF version of this book.

```
int width  = 400;
int height = 200;

OEImage image = new OEImage(width, height);

OEGraphMol mol = new OEGraphMol();
```

(continues on next page)

(continued from previous page)

```
oechem.OESmilesToMol(mol, "Cc1cccnc1/C=C/[C@H](C(=O)O)O");
oedepict.OEPrepareDepiction(mol);

OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, OEScale.AutoScale);
opts.SetMargins(10);
OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);

OEFont font = new OEFont(OEFontFamily.Default, OEFontStyle.Default, 12,
                         OEAlignment.Center, oechem.getOEDarkRed());

for (OE2DBondDisplay bdisp : disp.GetBondDisplays()) {
    OEBondBase bond = bdisp.GetBond();
    String hovertext = "bond idx=" + Integer.toString(bond.GetIdx());
    oedepict.OEDrawSVGHoverText(disp, bdisp, hovertext, font);
}
oedepict.OERenderMolecule("HoverBondText.svg", disp);
```

There is no image available for the PDF version of this book.

See also

- *OE2DMolDisplay* class
- *OE2DAtomDisplay* and *OE2DBondDisplay* classes
- *OEFont* class
- *OEDrawSVGToggleText* function

4.3.51 OEDrawSVGToggleText

```
bool OEDrawSVGToggleText(OE2DMolDisplay& disp, OE2DAtomDisplay* adisp,
                         const std::string& text, const OEFont& font)
bool OEDrawSVGToggleText(OE2DMolDisplay& disp, OE2DBondDisplay* bdisp,
                         const std::string& text, const OEFont& font)
```

Creates and interactive effect in `.svg` images. The given text will appear / disappear when clicking at the position defined by the given atom / bond display.

Note

This functionality is **only available** for `.svg` image format. In other image formats it has no effects.

The generated `svg` image should be included into and HTML page with the SVG MIME type.

```
<object data="<imagename>.svg" type="image/svg+xml"></object>
```

See also

- *Generating Interactive SVG Images* chapter

disp
The *OE2DMolDisplay* object that holds the data necessary to depict the molecule with which it is initialized.

adisp
The *OE2DAtomDisplay* object that defines where the text will appear.

bdisp
The *OE2DBondDisplay* object that defines where the text will appear.

text
The text that appears/disappears on mouse clicks.

font
The graphical properties of the font used to draw the text.

```
int width  = 400;
int height = 200;

OEImage image = new OEImage(width, height);

OEGraphMol mol = new OEGraphMol();
oechem.OESmilesToMol(mol, "Cc1cccnc1/C=C/[C@H](C(=O)O)O");
oedepict.OEPrepareDepiction(mol);

OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, OEScale.AutoScale);
opts.SetMargins(10);
OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);

OEFont font = new OEFont(OEFontFamily.Default, OEFontStyle.Default, 12,
                         OEAlignment.Center, oechem.getOEDarkRed());

for (OE2DAtomDisplay adisp : disp.GetAtomDisplays()) {
    OEAtomBase atom = adisp.GetAtom();
    String toggletext = "atom idx=" + Integer.toString(atom.GetIdx());
    oedepict.OEDrawSVGToggleText(disp, adisp, toggletext, font);
}
oedepict.OERenderMolecule("ToggleAtomText.svg", disp);
```

There is no image available for the PDF version of this book.

```
int width  = 400;
int height = 200;

OEImage image = new OEImage(width, height);

OEGraphMol mol = new OEGraphMol();
oechem.OESmilesToMol(mol, "Cc1cccnc1/C=C/[C@H](C(=O)O)O");
oedepict.OEPrepareDepiction(mol);

OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(width, height, OEScale.AutoScale);
opts.SetMargins(10);
```

(continues on next page)

(continued from previous page)

```
OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);

OEFont font = new OEFont(OEFontFamily.Default, OEFontStyle.Default, 12,
                         OEAlignment.Center, oechem.getOEDarkRed());

for (OE2DBondDisplay bdisp : disp.GetBondDisplays()) {
    OEBondBase bond = bdisp.GetBond();
    String toggletext = "bond idx=" + Integer.toString(bond.GetIdx());
    oedepict.OEDrawSVGToggleText(disp, bdisp, toggletext, font);
}
oedepict.OERenderMolecule("ToggleBondText.svg", disp);
```

There is no image available for the PDF version of this book.

See also

- *OE2DMolDisplay* class
- *OE2DAtomDisplay* and *OE2DBondDisplay* classes
- *OEFont* class
- *OEDrawSVGHoverText* function

4.3.52 OEDrawText

```
void OEDrawText(OEImageBase& image, const std::string& text,
                const OEFont& font, double padding = 5.0)
```

Draws the given text to the image respecting the font alignment.

text
The text that is going to be rendered into the given image.

font
The *OEFont* object that encapsulates properties that determine the display style of text.

padding
Defines the padding used on the left and right of the given cell as a percentage of the width of the cell.

See also

- *OEImageBase* class
- *OEFont* class
- *OEDrawTextToCenter* function

4.3.53 OEDrawTextToCenter

```
void OEDrawTextToCenter(OEImageBase& image, const std::string& text,
                        const OEFont& font)
```

Draws the given text to the center of the image.

See code snippet below and the corresponding image in *Figure: Example of using the OEDrawTextToCenter function*.

```
int width  = 100;
int height = 50;
OEImage image = new OEImage(width, height);

oedepict.OEDrawTextToCenter(image, "center", oedepict.getOEDefaultFont());
oedepict.OEDrawBorder(image, oedepict.getOELightGreyPen());

oedepict.OEWriteImage("OEDrawTextToCenter.png", image);
```



Fig. 352: **Example of using the OEDrawTextToCenter function**

See also

- *OEImageBase* class
- *OEFont* class
- *OEDrawBorder* function
- *OEDrawText* function

4.3.54 OEEstimateTextWidth

```
double OEEstimateTextWidth(const std::string& text, const OEFont& font)
```

Estimates the width of string depicted with a specific style.

text
: The string of which width is being estimated.

font
: The *OEFont* object that defines the style of the depiction.

See also

- *OEReduceFontSizeToFit* function
- *OEFont* class

4.3.55 OEGetAverageBondLengthInDispCoords

```
double OEGetAverageBondLengthInDispCoords(const OE2DMolDisplay& disp)
```

Returns the average bond length of the molecule in “display” coordinate system.

4.3.56 OEGetBuiltInHiddenSVGClass

```
const OESVGClass* OEGetBuiltInHiddenSVGClass(OEImageBase& image)
```

Returns an *OESVGClass* that have been previously added to the given image by calling the *OEAddBuiltInHiddenSVGClass* function.

See also

- *OESVGClass* class
- *OEImage* class
- *OEAddBuiltInHiddenSVGClass* function
- *OEAddBuiltInVisibleSVGClass* function
- *OEGetBuiltInVisibleSVGClass* function

4.3.57 OEGetBuiltInVisibleSVGClass

```
const OESVGClass* OEGetBuiltInVisibleSVGClass(OEImageBase& image)
```

Returns an *OESVGClass* that have been previously added to the given image by calling the *OEAddBuiltInVisibleSVGClass* function.

See also

- *OESVGClass* class
- *OEImage* class
- *OEAddBuiltInHiddenSVGClass* function
- *OEAddBuiltInVisibleSVGClass* function
- *OEGetBuiltInHiddenSVGClass* function

4.3.58 OEGetCenter

```
OE2DPoint OEGetCenter(const OEImageBase& image)
```

Returns the center of the image.

See also

- *OE2DPoint* class

4.3.59 OEGetColor

```
OESystem::OEColor OEGetColor(const OESystem::OEInterface& itf,
                             const std::string& name)
```

Returns the color initialized in the given interface.

itf
The interface in which the parameter is configured.

name
The name of the parameter.

See also

- *OEConfigureColor* function
- OEColor class

4.3.60 OEGetDefaultAtomColor

```
OESystem::OEColor OEGetDefaultAtomColor(unsigned int atomic,
                                        unsigned int style = OEAtomColorStyle::WhiteCPK)
```

Returns the default color associated with an atomic number for the given atom color style.

atomic
Atomic number.

style
This value has to be from the *OEAtomColorStyle* namespace.

See also

Appendix: Element coloring (CPK) chapter

4.3.61 OEGetDisplayCoords

```
OE2DPoint OEGetDisplayCoords(const OE2DMolDisplay& disp,
                             const OE2DPoint& p)
```

Converts a 2D point from the "molecule" 2D coordinate system" to "display" 2D coordinate system.

This is the reverse operation from *OEGetDisplayCoords*.

See also

- *OEGetMoleculeCoords* function

4.3.62 OEGetHighlightColor

```
OESystem::OEColor OEGetHighlightColor(const OESystem::OEInterface& itf,
                                      std::string name = OEHighlightParamName::Color)
```

Returns the highlight color initialized in the given interface.

itf
The interface in which the parameter is configured.

name
The name of the parameter.

See also

- *OEConfigureHighlightParams* function
- *OEConfigureHighlightColor* function
- OEColor class

4.3.63 OEGetHighlightStyle

```
unsigned int OEGetHighlightStyle(const OESystem::OEInterface& itf,
                                 std::string name = OEHighlightParamName::Style)
```

Returns the highlight style initialized in the given interface. The return value is taken from the *OEHighlightStyle* namespace.

itf
The interface in which the parameter is configured.

name
The name of the parameter.

See also

- *OEConfigureHighlightParams* function
- *OEConfigureHighlightStyle* function
- *OEHighlightStyle* namespace

4.3.64 OEGetImageGridNumColumns

```
unsigned int OEGetImageGridNumColumns(const OESystem::OEInterface& itf,
                                      std::string name =␣
↪OEImageGridParamName::NumColumns)
```

Returns the number of grid columns initialized in the given interface.

itf
The interface in which the parameter is configured.

name
The name of the parameter.

See also

- *OEConfigureImageGridNumColumns* function

4.3.65 OEGetImageGridNumRows

```
unsigned int OEGetImageGridNumRows(const OESystem::OEInterface& itf,
                                   std::string name = OEImageGridParamName::NumRows)
```

Returns the number of grid rows initialized in the given interface.

itf
The interface in which the parameter is configured.

name
The name of the parameter.

See also

- *OEConfigureImageGridNumRows* function

4.3.66 OEGetImageHeight

```
double OEGetImageHeight(const OESystem::OEInterface& itf,
                        std::string name = OEImageParamName::Height)
```

Returns the height of the image initialized in the given interface.

itf
The interface in which the parameter is configured.

name
The name of the parameter.

See also

- *OEConfigureImageHeight* function

4.3.67 OEGetImageWidth

```
double OEGetImageWidth(const OESystem::OEInterface& itf,
                       std::string name = OEImageParamName::Width)
```

Returns the width of the image initialized in the given interface.

itf
The interface in which the parameter is configured.

name
The name of the parameter.

See also

- *OEConfigureImageWidth* function

4.3.68 OEGetLabelPosition

Returns the position used to add a label to an image or molecule display.

Table 49: **The overloaded versions of the OEGetLabelPosition function**

Link	Description
OEGetLabelPosition(disp, label, match)	getting label position related to atoms of a match
OEGetLabelPosition(disp, label, abset)	getting label position related to set of atoms
OEGetLabelPosition(disp, label, atompred)	getting label position related to atoms specified by a predicate

Note

When adding label associated with atoms it is recommended to use the *OEAddLabel* along with the *OEAddHighlighting* function for visual clarity.

```
OE2DPoint OEGetLabelPosition(const OE2DMolDisplay& disp,
                             const OEHighlightLabel& label,
                             const OEChem::OEAtomBondSet& abset)
```

disp
The *OE2DMolDisplay* object on which the label is going to be positioned.

label
The *OEHighlightLabel* object that stores properties that determine the styles of the label along with the text of the label itself.

abset
The label is positioned based on the atoms stored in the OEAtomBondSet object.

```
OE2DPoint OEGetLabelPosition(const OE2DMolDisplay& disp,
                             const OEHighlightLabel& label,
                             const OEChem::OEMatchBase& match)
```

Adds label to atoms of an OEMatchBase object that can be initialized by substructure search or maximum common substructure search.

disp The *OE2DMolDisplay* object on which the label is going to be positioned.

label
The *OEHighlightLabel* object that stores properties that determine the styles of the label along with the text of the label itself.

match
The label is positioned based on the target atoms of the OEMatchBase object.

```
OE2DPoint OEGetLabelPosition(const OE2DMolDisplay& disp,
                             const OEHighlightLabel& label,
                             const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&␣
↪atompred)
```

disp
The *OE2DMolDisplay* object on which the label is going to be positioned.

label
The *OEHighlightLabel* object that stores properties that determine the styles of the label along with the text of the label itself.

atompred
The label is positioned based on the atoms for which the given atom predicate returns *true*.

4.3.69 OEGetMoleculeCoords

```
OE2DPoint OEGetMoleculeCoords(const OE2DMolDisplay& disp,
                              const OE2DPoint& p)
```

Converts a 2D point from the "display" 2D coordinate system" to "molecule" 2D coordinate system.

This is the reverse operation from *OEGetMoleculeCoords*.

See also

- *OEGetMoleculeCoords* function

4.3.70 OEGetMoleculeScale

```
double OEGetMoleculeScale(const OEChem::OEMolBase& mol,
                          const OE2DMolDisplayOptions& opts)
```

Estimates the scaling factor of the molecule depiction based on the parameters stored in given *OE2DMolDisplayOptions* object.

mol
The molecule of which scaling factor is determined.

opts
The *OE2DMolDisplayOptions* object that stores the parameters that determine the scaling factor (such as width and height).

See also

- *Controlling the Size of Depicted Molecules* section

4.3.71 OEGetMultiPageOrientation

```
unsigned int OEGetMultiPageOrientation(const OESystem::OEInterface& itf,
                                       std::string name =␣
↪OEMultiPageParamName::PageOrientation)
```

Returns the page orientation initialized in the given interface. The return value is taken from the *OEPageOrientation* namespace.

itf
The interface in which the parameter is configured.

name
The name of the parameter.

See also

- *OEConfigureMultiPageOrientation* function

4.3.72 OEGetMultiPageSize

```
unsigned int OEGetMultiPageSize(const OESystem::OEInterface& itf,
                                std::string name = OEMultiPageParamName::PageSize)
```

Returns the page orientation initialized in the given interface. The return value is taken from the *OEPageSize* namespace.

itf
The interface in which the parameter is configured.

name
The name of the parameter.

See also

- *OEConfigureMultiPageSize* function

4.3.73 OEGetNearbyAtom

```
OENearestAtom OEGetNearbyAtom(const OE2DMolDisplay& disp,
                              const OE2DPoint& point,
                              double radiusscale = 1.0)
```

Returns an *OENearestAtom* object that stores a pointer of the atom which is depicted nearest to a given 2D display coordinates. By default, only atoms that are no further than half bond away from the given point will be considered.

disp
The *OE2DMolDisplay* object of which atoms are being considered. Only atoms of which atom display is visible are examined (*OEAtomDisplayBase.IsVisible*)

point
Specifies the 2D display coordinates.

radiusscale
The multiplier that can be used to increase or to decrease the radius around the given point to be considered. This value has to be in a range of $[0.25, 2.0]$.

Note

If there is no atom close to the given display coordinates, then the *OENearestAtom.IsValid* method returns *false*.

See also

- *OENearestAtom* class
- *OEGetNearestAtom* function
- *OEGetNearbyBond* function

4.3.74 OEGetNearbyBond

```
OENearestBond OEGetNearbyBond(const OE2DMolDisplay& disp,
                              const OE2DPoint& point,
                              double radiusscale = 1.0)
```

Returns an *OENearestAtom* object that stores a pointer of the bond which is depicted nearest to a given 2D display coordinates. By default, only bonds that are no further than half bond away from the given point will be considered.

disp
: The *OE2DMolDisplay* object of which bonds are being considered. Only bonds of which bond display is visible are examined (*OEBondDisplayBase.IsVisible*)

point
: Specifies the 2D display coordinates.

radiusscale
: The multiplier that can be used to increase or to decrease the radius around the given point to be considered. This value has to be in a range of $[0.25, 2.0]$.

Note

If there is no bond close to the given display coordinates, then the *OENearestBond.IsValid* method returns *false*.

See also

- *OENearestBond* class
- *OEGetNearestBond* function
- *OEGetNearbyAtom* function

4.3.75 OEGetNearestAtom

```
OENearestAtom OEGetNearestAtom(const OE2DMolDisplay& disp,
                               const OE2DPoint& point)
```

Returns an *OENearestAtom* object that stores a pointer of the atom which is depicted nearest to a given 2D display coordinates.

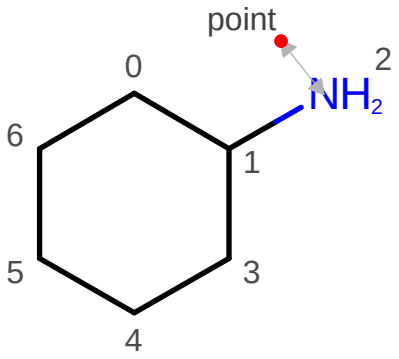


Fig. 353: **Example of using the OEGetNearestAtom function to determine the nearest atom to a specific 2D point on the canvas**

disp
The *OE2DMolDisplay* object of which atoms are being considered. Only atoms of which atom display is visible are examined (`OEAtomDisplayBase.IsVisible`)

point
Specifies the 2D display coordinates.

See also

- *OENearestAtom* class
- `OEGetNearbyAtom` function

4.3.76 OEGetNearestBond

```
OENearestBond OEGetNearestBond(const OE2DMolDisplay& disp,
                               const OE2DPoint& point)
```

Returns an *OENearestBond* object that stores a pointer of the bond which is depicted nearest to a given 2D display coordinates.

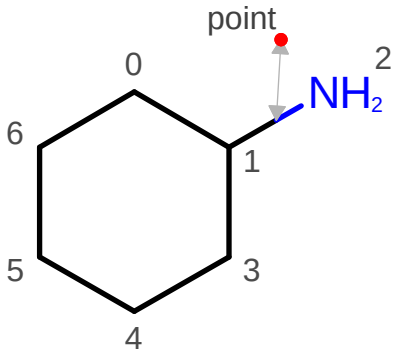


Fig. 354: **Example of using the OEGetNearestBond function to determine the nearest bond to a specific 2D point on the canvas**

disp
The *OE2DMolDisplay* object of which bonds are being considered. Only bonds of which bond display is visible are examined (*OEBondDisplayBase.IsVisible*)

point
Specifies the 2D display coordinates.

See also

- *OENearestBond* class
- *OEGetNearbyBond* function

4.3.77 OEGetPageHeight

```
double OEGetPageHeight(unsigned int size)
```

Returns the page height for the given size.

size
This value has to be from the *OEPageSize* namespace.

See also

- *OEPageSize* namespace
- *OEGetPageWidth* function

4.3.78 OEGetPageWidth

```
double OEGetPageWidth(unsigned int size)
```

Returns the page width for the given size.

size
This value has to be from the *OEPageSize* namespace.

See also

- *OEPageSize* namespace
- *OEGetPageHeight* function

4.3.79 OEGetSupportedImageFileExtensions

```
OESystem::OEIterBase<const std::string>* OEGetSupportedImageFileExtensions()
```

Returns an iterator over the file extensions of the supported image file formats.

Graphics File Format	File Extension
PNG (Portable Network Graphics)	`.png`
SVG (Scalable Vector Graphics)	`.svg`
bare SVG (with no header)	`.bsvg`
Postscript	`.ps`
Encapsulated PostScript	`.eps`
PDF (Portable Document Format)	`.pdf`

4.3.80 OEGetSupportedMultiPageImageFileExtensions

```
OESystem::OEIterBase<const std::string>* OEGetSupportedMultiPageImageFileExtensions()
```

Returns an iterator over the file extensions of the supported multi-page image file formats.

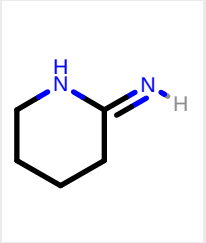
Multi-page Image File Format	File Extension
Postscript	`.ps`
PDF (Portable Document Format)	`.pdf`

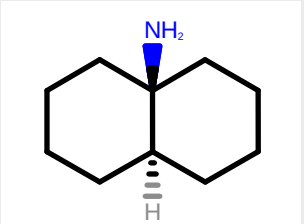
4.3.81 OEHasDepictionHydrogens

```
bool OEHasDepictionHydrogens(const OEChem::OEAtomBase *)
```

Determines whether the specified heavy atom has a "depiction" hydrogen. Depiction hydrogens are hydrogens that need to be explicitly drawn in a 2D depiction to faithfully represent tetrahedral atom stereochemistry or cis/trans bond stereochemistry.

Table 50: **Example of molecules with "depiction" hydrogen(s)**





See also

- *OEAddDepictionHydrogens* function

4.3.82 OEIsRegisteredImageFile

```
bool OEIsRegisteredImageFile(const std::string& ext)
```

Returns whether or not an image file format is registered (*i.e.* supported) with the given extension. The following snippet of code demonstrates how to check whether a file name is a supported file format.

```
String filename = argv[0];
String ext = oechem.OEGetFileExtension(filename);
if (!oedepict.OEIsRegisteredImageFile(ext))
    oechem.OEThrow.Fatal("Unknown image type!");
```

4.3.83 OEIsRegisteredMultiPageImageFile

```
bool OEIsRegisteredMultiPageImageFile(const std::string& ext)
```

Returns whether or not a multi-page image file format is registered (*i.e.* supported) with the given extension.

4.3.84 OEIsValidSVGClassName

```
bool OEIsValidSVGClassName(const std::string& name)
```

Returns whether the given string can be used as an SVG class name.

The rules of **valid** SVG classes names (*OESVGClass*) in **OEDepict TK** are:

- It must start with an alphabetic character.
- It must not contain any space characters.
- It may contain any alphanumeric characters, `-` (hyphen) and `_` (underscore).
- The size of the SVG class name should not exceed `64` characters.

> **Note**
>
> SVG class names starting with `oedepict` are reserved by **OEDepict TK** for internal usage.

4.3.85 OEIsValidSVGGroupId

```
bool OEIsValidSVGGroupId(const std::string& id)
```

Returns whether the given string can be used as an SVG group identifier.

The rules of **valid** SVG group ids (*OESVGClass*) in **OEDepict TK** are:

- It must start with an alphabetic character.
- It must not contain any space characters.
- It may contain alphanumeric character and `-` (hyphen), `_` (underscore), `;` (semi-colon) or `.` (period).
- The size of the SVG group id should not exceed `64` characters.

> **Note**
>
> The group id must be **unique** amongst all the ids in the document. This condition is not checked by the *OEIsValidSVGGroupId* function.

4.3.86 OELengthenVector

```
OE2DPoint OELengthenVector(const OE2DPoint& vect, double length)
```

Lengthen the given vector.

See also

- *OE2DPoint* class
- *OERotateVector* function

4.3.87 OEOffsetMolDisplay

```
void OEOffsetMolDisplay(OE2DMolDisplay& disp, const OE2DPoint& offset)
void OEOffsetMolDisplay(OE2DMolDisplay& disp,
                        const OE2DMolDisplay& refdisp)
```

When an *OE2DMolDisplay* object is created, the depicted molecule is positioned in the middle of the *canvas*. The *OEOffsetMolDisplay* function can be used to translate the depicted molecule either by specifying an *offset* vector or another *OE2DMolDisplay* object that is used as a reference that defines the translation.

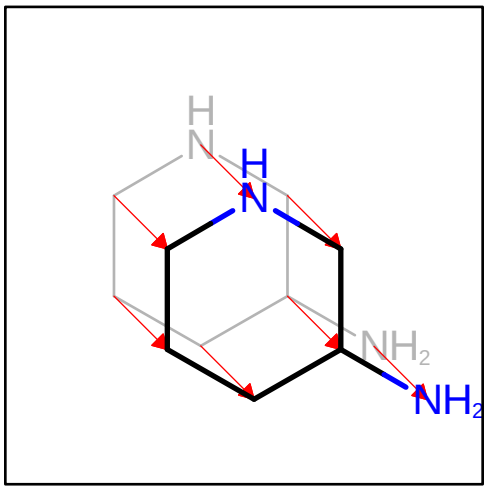


Fig. 355: **Example of translating the depiction of a molecule by using offset vector (red arrows)**

4.3.88 OEPrepareAlignedDepiction

Table 51: **The overloaded versions of the OEPrepareAlignedDepiction function**

Link	Description
OEPrepareAlignedDepiction(fitmol, refmol, match, clearcoords, suppressH)	aligns a molecule to the reference
OEPrepareAlignedDepiction(fitmol, refmol, match, opts)	aligns a molecule to the reference according to options
OEPrepareAlignedDepiction(fitmol, mcss, opts)	aligns a molecule to the reference based on maximum common substructure search
OEPrepareAlignedDepiction(fitmol, subsrc, opts)	aligns a molecule to the reference based on substructure search
OEPrepareAlignedDepiction(fitmol, refmol, matches, opts)	aligns a molecule to the reference based on matches

```
bool OEPrepareAlignedDepiction(OEChem::OEMolBase& fitmol,
                               const OEChem::OEMolBase& refmol,
                               const OEChem::OEMatchBase& match,
                               bool clearCoords = true,
                               bool suppressH = true)
```

Generates the 2D coordinates of a molecule by maximizing its overlap with a reference 2D molecule based on the given match between the two molecules.

fitmol
: The molecule being fitted to be aligned to the reference molecule.

refmol
: The reference molecule of which 2D coordinates are remain unchanged.

match
: The match determines the atoms of the *fitmol* that will be aligned to the atoms of the *refmol*. The match can be generated by hand, or with any of **OEChem TK**'s matching algorithms such as OESubSearch or OEMCSSearch.

clearcoords
: If *true*, than the 2D coordinates of the fitted molecule are re-calculated by calling the `OEPrepareDepiction` function. After generating the coordinates, the alignment between the two molecules is maximized by rotating the single bonds of the fitted molecule while avoiding atom clashes. See example (C) in *Figure: Example of the molecule alignment*.

  If *clearcoords* is *false* and the *fitmol* has 2D coordinates than these coordinates are kept. In this case only 2D rotation and translation of the fitted molecule is allowed (*i.e.* no rotations around bonds) in order to align it to the reference molecule. See example (B) in *Figure: Example of the molecule alignment*.

suppressH
: If true, than explicit hydrogens are suppressed in the input molecule by calling the `OESuppressNonDepictionHydrogens` function.

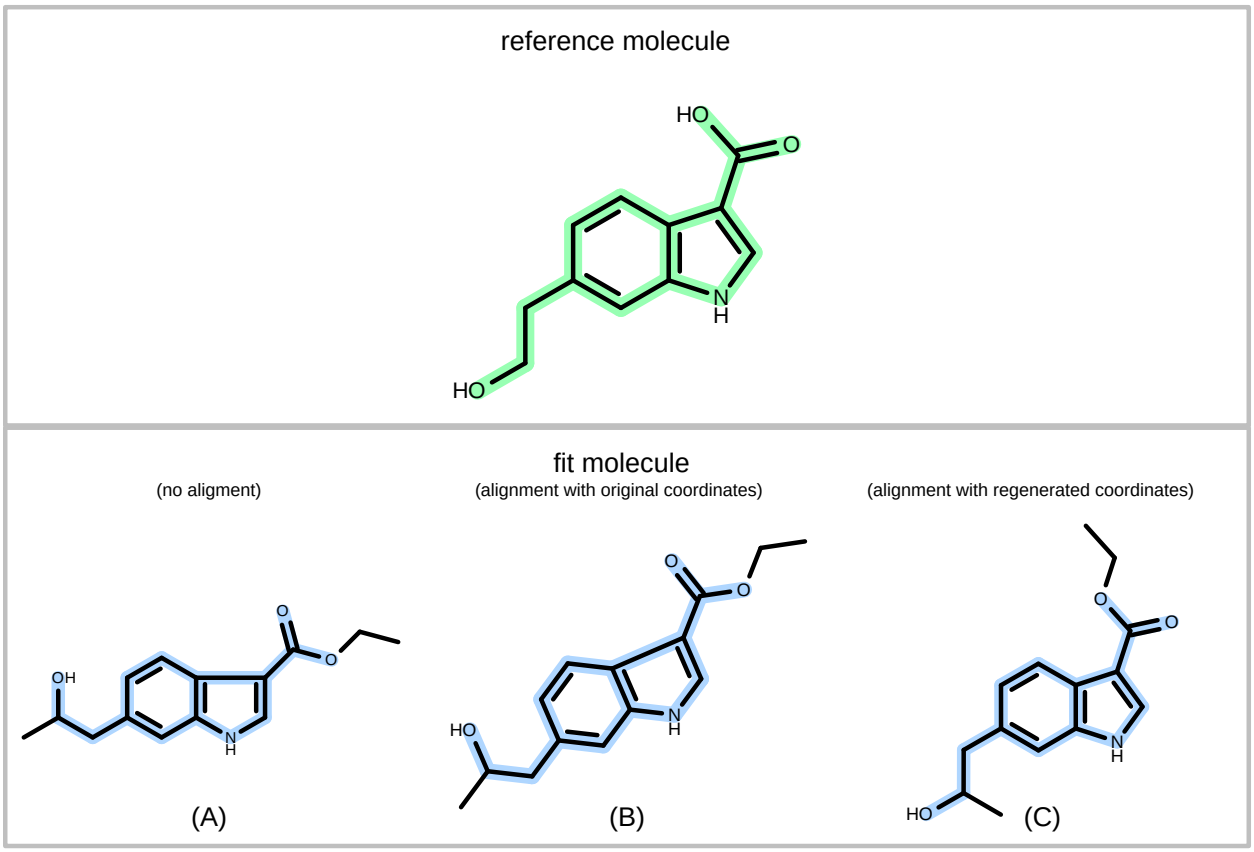


Fig. 356: **Example of the molecule alignment**
(A) molecule with no alignment; (B) aligning the fit molecule to the reference molecule by allowing only 2D rotations (original coordinates are kept); (C) aligning the fit molecule to the reference by allowing rotation around single bonds (coordinates are regenerate)

See also

- *Molecule Alignment* chapter

```
bool OEPrepareAlignedDepiction(OEChem::OEMolBase& fitmol,
                               const OEChem::OEMolBase& refmol,
                               const OEChem::OEMatchBase& match,
                               const OEAlignmentOptions& opts)
```

Generates the 2D coordinates of a molecule by maximizing its overlap with a reference 2D molecule based on the given match between the two molecules.

fitmol
The molecule being fitted to be aligned to the reference molecule.

refmol
The reference molecule of which 2D coordinates are remain unchanged.

match
The match determines the atoms of the *fitmol* that will be aligned to the atoms of the *refmol*. The match can be generated by hand, or with any of **OEChem TK**'s matching algorithms such as OESubSearch or OEMCSSearch.

opts
The *OEAlignmentOptions* object that stores properties that influence how the molecule alignment is performed.

See also

- *Molecule Alignment* chapter

```
OEAlignmentResult OEPrepareAlignedDepiction(OEChem::OEMolBase& fitmol,
                                            const OEChem::OEMCSSearch& mcss,
                                            const OEAlignmentOptions& opts =␣
↪OEAlignmentOptions())
```

Generates the 2D coordinates of a molecule by maximizing its overlap based on the maximum common substructure initialized with a reference 2D molecule.

fitmol
The molecule being fitted to be aligned to the reference molecule.

mcss
The OEMCSSearch object stores the reference molecule. The molecule alignments are performed based on the maximum common substructure(s) detected between the fit and the reference molecule.

opts
The *OEAlignmentOptions* object that stores properties that influence how the molecule alignment is performed.

Warning

The reference molecule has to have 2D coordinates prior to initializing the OEMCSSearch object.

See also

- OEMCSSearch class in the **OEChem TK** manual
- `Listing 1` example code in *Molecule Alignment* chapter

```
OEAlignmentResult OEPrepareAlignedDepiction(OEChem::OEMolBase& fitmol,
                                            const OEChem::OESubSearch& subsrc,
                                            const OEAlignmentOptions& opts =␣
↪OEAlignmentOptions())
```

Generates the 2D coordinates of a molecule by maximizing its overlap based on substructure search initialized with a reference 2D molecule.

In case of a successful alignment, the returned *OEAlignmentResult* object stores the match between fit and reference molecules.

fitmol
The molecule being fitted to be aligned to the reference molecule.

subsrc
The OESubSearch object stores the reference molecule. The molecule alignments are performed based on the substructure(s) detected between the fit and the reference molecule.

opts
The *OEAlignmentOptions* object that stores properties that influence how the molecule alignment is performed.

Warning

The reference molecule has to have 2D coordinates prior to initializing the OESubSearch object.

See also

- OESubSearch class in the **OEChem TK** manual
- `Listing 2` code example in *Molecule Alignment* chapter

```
OEAlignmentResult OEPrepareAlignedDepiction(OEChem::OEMolBase& fitmol,
                                            const OEChem::OEMolBase& refmol,
                                            OESystem::OEIter<const OEChem::OEMatchBase>&␣
↪miter,
                                            const OEAlignmentOptions& opts =␣
↪OEAlignmentOptions())
OEAlignmentResult OEPrepareAlignedDepiction(OEChem::OEMolBase& fitmol,
                                            const OEChem::OEMolBase& refmol,
                                            OESystem::OEIterBase<const␣
↪OEChem::OEMatchBase>* miter,
                                            const OEAlignmentOptions& opts =␣
↪OEAlignmentOptions())
```

Generates the 2D coordinates of a molecule by maximizing its overlap with a reference 2D molecule based on the given matches between the two molecules.

fitmol
The molecule being fitted to be aligned to the reference molecule.

refmol
The reference molecule of which 2D coordinates are remain unchanged.

miter
The iterator that stores the matches on which the molecule alignment is based on.

opts
The *OEAlignmentOptions* object that stores properties that influence how the molecule alignment is performed.

See also

- *Molecule Alignment* chapter

4.3.89 OEPrepareDepiction

Table 52: **The overloaded versions of the OEPrepareDepiction function**

Link	Description
OEPrepareDepiction(mol, clearcords, suppressH)	prepare depiction
OEPrepareDepiction(mol, opts)	prepare depiction according to options

```
bool OEPrepareDepiction(OEChem::OEMolBase& mol, bool clearcoords = false,
                        bool suppressH = true)
```

Prepares a molecule before depiction that involves suppressing or adding explicit hydrogens, perceiving atom and bond stereo information and calculating 2D coordinates.

mol
The molecule being manipulated.

clearcoords
If true, than the 2D coordinates of the molecule are re-calculated. If false and the molecule has 2D coordinates than these coordinates are kept.

If the molecule has no 2D coordinates, then the 2D coordinates are always generated by calling the `OEDepictCoordinates` function regardless of the value of the *clearcoords* parameter.

suppressH
If true, than explicit hydrogens are suppressed in the input molecule by calling the `OESuppressNonDepictionHydrogens` function. See example in *Figure: Example of the molecule depictions with kept and suppressed hydrogens*.

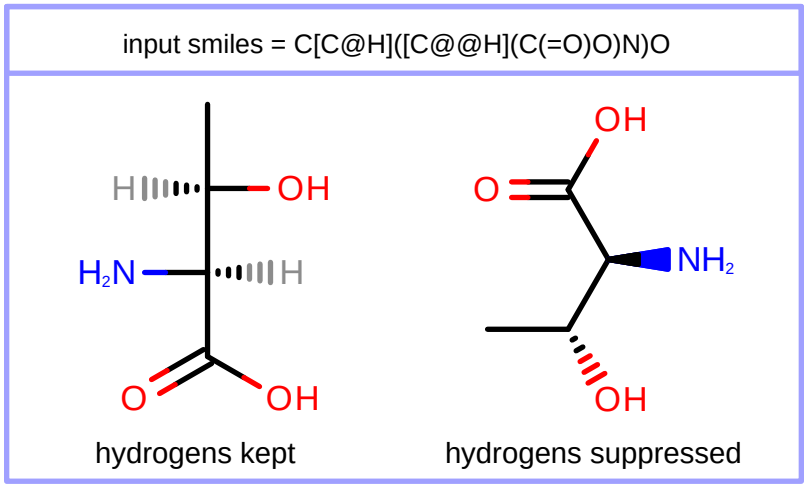


Fig. 357: **Example of the molecule depictions with kept and suppressed hydrogens**

Note

The *OEPrepareDepiction* function generates canonical 2D coordinates (when the *clearcoords* parameter is true) i.e. the layout of the same molecules will be identical regardless of their atom and bond orderings.

The following code shows how to generate non-canonical 2D coordinates:

```
private static boolean prepareDepiction(OEMolBase mol, boolean clearcoords,␣
↪boolean suppressH) {

    oechem.OESetDimensionFromCoords(mol);
    oechem.OEPerceiveChiral(mol);
    if (mol.GetDimension() != 2 || clearcoords) {
        if (mol.GetDimension() == 3) {
            oechem.OE3DToBondStereo(mol);
            oechem.OE3DToAtomStereo(mol);
        }
        if (suppressH)
            oechem.OESuppressHydrogens(mol);
        oedepict.OEAddDepictionHydrogens(mol);

        oedepict.OEDepictCoordinates(mol);
        oechem.OEMDLPerceiveBondStereo(mol);
    }
    mol.SetDimension(2);
    return true;
}
private static boolean prepareDepiction(OEMolBase mol, boolean clearcoords) {
    return prepareDepiction(mol, clearcoords, true);
}
private static boolean prepareDepiction(OEMolBase mol) {
    return prepareDepiction(mol, false, true);
}
```

See also

- *OEAddDepictionHydrogens* function
- *OEDepictCoordinates* function

```
bool OEPrepareDepiction(OEChem::OEMolBase& mol,
                        const OEPrepareDepictionOptions& opts)
```

Prepares a molecule before depiction.

mol
: The molecule being manipulated.

opts
: The *OEPrepareDepictionOptions* object that stores properties that influence how the molecule is prepared.

4.3.90 OEPrepareMultiAlignedDepiction

```
bool OEPrepareMultiAlignedDepiction(OEChem::OEMolBase& fitmol,
                                    const OEChem::OEMolBase& refmol,
                                    OESystem::OEIter<OEChem::OEMatchBase>& miter)
bool OEPrepareMultiAlignedDepiction(OEChem::OEMolBase& fitmol,
                                    const OEChem::OEMolBase& refmol,
                                    OESystem::OEIterBase<OEChem::OEMatchBase>* miter)
```

Generates the 2D coordinates of a molecule by maximizing its overlay with a reference 2D molecule based on multiple overlaps between the two molecules.

fitmol
The molecule being fitted to be aligned to the reference molecule.

refmol
The reference molecule of which 2D coordinates are remain unchanged.

miter
The iterator that stores the matches on which the molecule alignment is based on.

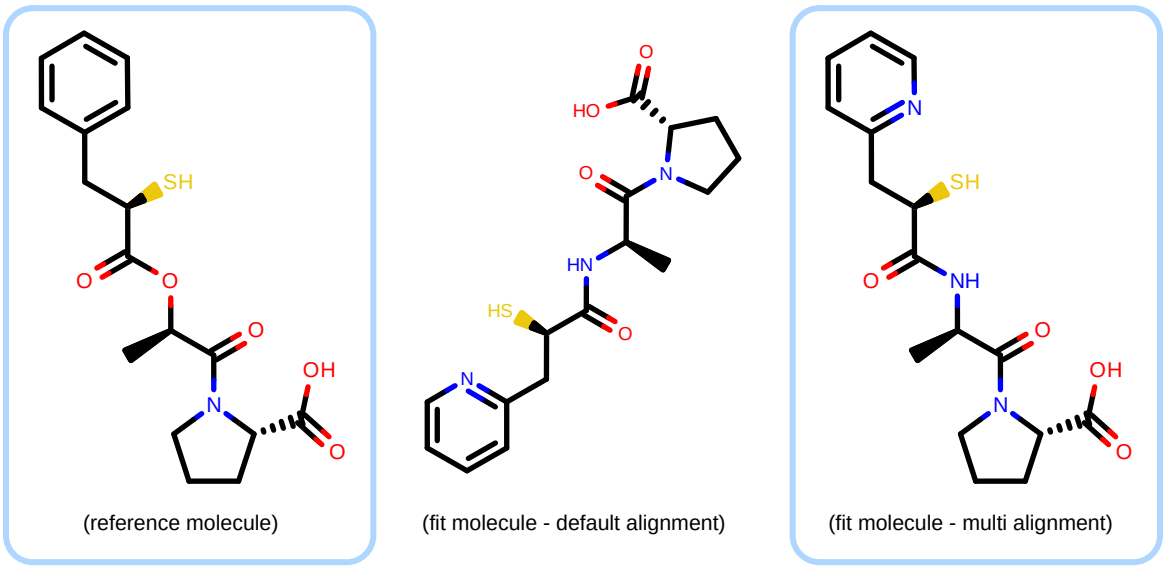
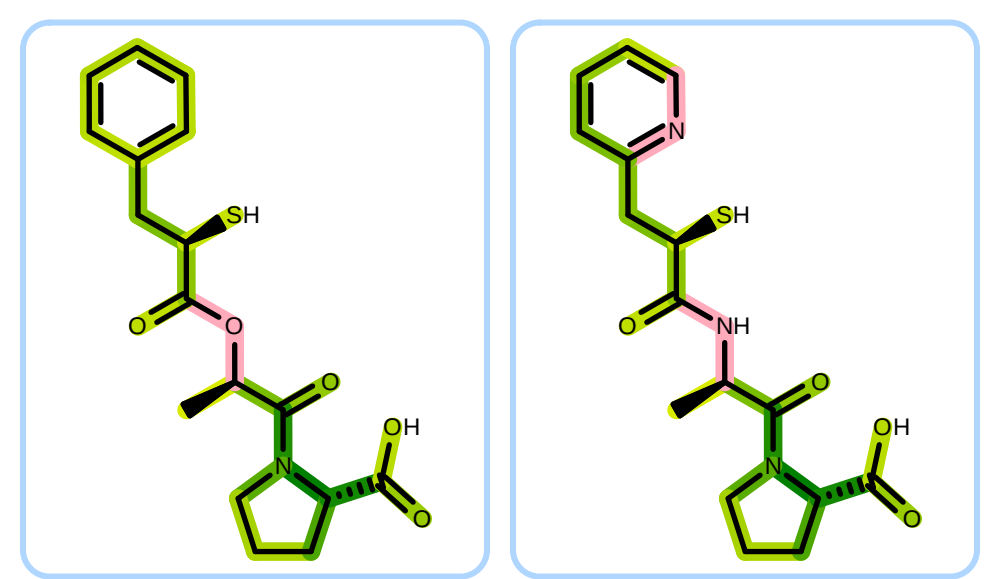


Fig. 358: **Example of the molecule alignment based on multiple fragments**
(left) - depiction of a reference molecule (middle) - default depiction of a fit molecule (right) - depiction of the fit molecule aligned to the reference based on multiple fragments

Hint

The overlap between two molecules can be generated by the `OEGetFPOverlap` function of **GraphSim TK**. The `OEGetFPOverlap` returns matched fragments of two molecules that are considered equivalent based on a specific fingerprint type.



The Python script that visualizes molecule similarity based on fingerprints can be downloaded from the OpenEye Python Cookbook

See also

- *Molecule Alignment* chapter
- Fingerprint Overlap chapter in **GraphSim TK**
- `OEPrepareAlignedDepiction` function
- `Listing 3` example code in *Molecule Alignment* chapter

4.3.91 OEReduceFontSizeToFit

```
unsigned int OEReduceFontSizeToFit(const std::string& text,
                                   const OEFont& font,
                                   unsigned int minfontsize,
                                   double maxwidth)
```

The `OEReduceFontSizeToFit` function gradually reduces the font size until the specified string can be drawn without exceeding the width limit.

text
The string for which the font size is calculated.

font
The *OEFont* object that defines the style of the depiction.

minfontsize
The minimum font size that can be returned.

maxwidth

The width that can not be exceeded when drawing the string.

See also

- *OEEstimateTextWidth* function
- *OEFont* class

4.3.92 OERegisterImageFile

```
bool OERegisterImageFile(const OEImageFileCreatorBase &)
```

Registers an *OEImageFileCreatorBase* object that handles a specific image file type.

See also

- *OEImageFileCreatorBase* class
- *OEIsRegisteredImageFile* function
- *OEIsRegisteredMultiPageImageFile* function

4.3.93 OERenderMolecule

Table 53: **The overloaded versions of the OERenderMolecule function**

Link	Description
OERenderMolecule(image, disp)	renders a molecule display into an image
OERenderMolecule(image, mol)	renders molecule into an image using the **default** depiction style
OERenderMolecule(filename, disp)	writes a molecule display into a file
OERenderMolecule(filename, mol)	writes a molecule into a file using the **default** depiction style
OERenderMolecule(stream, ext, disp)	writes a molecule display into an output stream
OERenderMolecule(stream, ext, mol)	writes a molecule into an output stream using the **default** depiction style

Hint

Setting the *clearbackground* option to false, allows the generation of images with transparent background.

clearbackground[Page 614, 1]	effect
true (default)	clears the image by drawing a rectangle underneath the molecule with the background color[2] before molecule rendering
false	does **not** clear the image, enabling the generation of transparent images (for example PNG, SVG)

See the Generating Transparent PNG Python cookbook example.

```
bool OERenderMolecule(OEImageBase& image, const OE2DMolDisplay& disp,
                      bool clearbackground = true)
```

Renders the molecule display into an image.

image
The image in which the molecule is drawn.

disp
The *OE2DMolDisplay* object that holds the data necessary to depict the molecule with which it is initialized.

clearbackground
Determines whether or not the image is cleared (by calling *OEImageBase.Clear* method) prior to rendering the molecule; as a result, by default a white rectangle will be drawn underneath the molecule.

Example:

```
oedepict.OEPrepareDepiction(mol);

OEImage image = new OEImage(200.0, 200.0);
OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(image.GetWidth(), image.
↪GetHeight(), OEScale.AutoScale);
OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
oedepict.OERenderMolecule(image, disp);
```

See also

- *OEImageBase* class
- *OE2DMolDisplay* class

```
bool OERenderMolecule(OEImageBase& image,
                      const OEChem::OEMolBase& mol,
                      bool clearbackground = true)
```

Renders a molecule into an image using the **default** depiction style.

image
The image in which the molecule is drawn.

mol
The molecule being rendered has to have valid 2D coordinates before calling the *OERenderMolecule* function.

clearbackground
Determines whether or not the image is cleared (by calling *OEImageBase.Clear* method) prior to rendering the molecule; as a result, by default a white rectangle will be drawn underneath the molecule.

Example:

```
oedepict.OEPrepareDepiction(mol);

OEImage image = new OEImage(200.0, 200.0);
oedepict.OERenderMolecule(image, mol);
```

[1] option of the *OERenderMolecule* function
[2] background color can be set with the *OE2DMolDisplayOptions.SetBackgroundColor* method (default is `OEWhite`)

See also

- *OEImageBase* class

```
bool OERenderMolecule(const std::string& filename,
                      const OE2DMolDisplay& disp,
                      bool clearbackground = true)
```

Writes the molecule display into a file.

filename
The name of the file into which the molecule display is being written. The extension of the filename determine the type of the image.

disp
The *OE2DMolDisplay* object that holds the data necessary to depict the molecule with which it is initialized.

clearbackground
Determines whether or not the image is cleared (by calling *OEImageBase.Clear* method) prior to rendering the molecule; as a result, by default a white rectangle will be drawn underneath the molecule.

Example:

```
oedepict.OEPrepareDepiction(mol);

OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(200.0, 200.0, OEScale.AutoScale);
OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);
oedepict.OERenderMolecule("OERenderMolecule-fname-disp.png", disp);
```

See also

- *OE2DMolDisplay* class

```
bool OERenderMolecule(const std::string& filename,
                      const OEChem::OEMolBase& mol,
                      bool clearbackground = true)
```

Writes a molecule into a file using the **default** depiction style.

filename
The name of the file into which the molecule is written. The extension of the filename determine the type of the image.

mol
The molecule being rendered has to have valid 2D coordinates before calling the *OERenderMolecule* function.

clearbackground
Determines whether or not the image is cleared (by calling *OEImageBase.Clear* method) prior to rendering the molecule; as a result, by default a white rectangle will be drawn underneath the molecule.

Example:

```
oedepict.OEPrepareDepiction(mol);

oedepict.OERenderMolecule("OERenderMolecule-fname-mol.png", mol);
```

```
bool OERenderMolecule(OEPlatform::oeostream& os,
                      const std::string& extension,
                      const OE2DMolDisplay& disp,
                      bool clearbackground = true)
```

Writes the molecule display into a stream.

os
The stream into which the molecule display is being written.

ext
The extension which determine the type of the image.

disp
The *OE2DMolDisplay* object that holds the data necessary to depict the molecule with which it is initialized.

clearbackground
Determines whether or not the image is cleared (by calling *OEImageBase.Clear* method) prior to rendering the molecule; as a result, by default a white rectangle will be drawn underneath the molecule.

Example:

```
oedepict.OEPrepareDepiction(mol);

OE2DMolDisplayOptions opts = new OE2DMolDisplayOptions(200.0, 200.0, OEScale.AutoScale);
OE2DMolDisplay disp = new OE2DMolDisplay(mol, opts);

oeofstream ofs = new oeofstream("OERenderMolecule-stream-disp.png");
oedepict.OERenderMolecule(ofs, "png", disp);
ofs.close();
```

See also

- oeostream class in the **OEChem TK** manual
- *OE2DMolDisplay* class

```
bool OERenderMolecule(OEPlatform::oeostream& os,
                      const std::string& extension,
                      const OEChem::OEMolBase& mol,
                      bool clearbackground = true)
```

Writes a molecule into a stream using the **default** depiction style.

os
The stream into which the molecule is being written.

ext
The extension which determine the type of the image.

mol
The molecule being rendered has to have valid 2D coordinates before calling the *OERenderMolecule* function.

clearbackground
Determines whether or not the image is cleared (by calling *OEImageBase.Clear* method) prior to rendering the molecule; as a result, by default a white rectangle will be drawn underneath the molecule.

Example:

```
oedepict.OEPrepareDepiction(mol);

oeofstream ofs = new oeofstream("OERenderMolecule-stream-mol.png");
oedepict.OERenderMolecule(ofs, "png", mol);
ofs.close();
```

See also

- oeostream class in the **OEChem TK** manual

4.3.94 OERenderMoleculeToString

Table 54: **The overloaded versions of the OERenderMoleculeToString function**

Link	Description
OERenderMoleculeToString(ext, disp)	writes a molecule display into a string (byte array)
OERenderMoleculeToString(ext, mol)	writes a molecule into a string (byte array) using the **default** depiction style

```
std::string OERenderMoleculeToString(const std::string& ext,
                                     const OE2DMolDisplay& disp,
                                     bool clearbackground = true)
```

Writes a molecule display into a string.

ext
The extension which determine the type of the image.

disp
The *OE2DMolDisplay* object that holds the data necessary to depict the molecule with which it is initialized.

clearbackground
Determines whether or not the image is cleared (by calling *OEImageBase.Clear* method) prior to the molecule rendering.

See also

- *OEIsRegisteredImageFile* function

```
std::string OERenderMoleculeToString(const std::string& ext,
                                     const OEChem::OEMolBase& mol,
                                     bool clearbackground = true)
```

Writes a molecule into a string using the **default** depiction style.

ext
The extension which determine the type of the image.

mol
The molecule being rendered.

clearbackground
Determines whether or not the image is cleared (by calling *OEImageBase.Clear* method) prior to the molecule rendering.

Example:

```
OEGraphMol mol = new OEGraphMol();
oechem.OESmilesToMol(mol, "c1ccccc1");
oedepict.OEPrepareDepiction(mol);

byte [] bytes = oedepict.OERenderMoleculeToByteArray("svg", mol);
```

See also

- *OEIsRegisteredImageFile* function

4.3.95 OERotateVector

```
OE2DPoint OERotateVector(const OE2DPoint& vect, double degree)
```

Rotates the vector with the given degree.

See also

- *OE2DPoint* class
- *OELengthenVector* function

4.3.96 OESetup2DMolDisplayOptions

```
bool OESetup2DMolDisplayOptions(OE2DMolDisplayOptions& opts,
                                const OESystem::OEInterface& itf)
```

Initializes an *OE2DMolDisplayOptions* object from the parameters of a given interface.

See also

- *OE2DMolDisplayOptions* class
- *OEConfigure2DMolDisplayOptions* function

4.3.97 OESetupPrepareDepictionOptions

```
bool OESetupPrepareDepictionOptions(OEPrepareDepictionOptions& opts,
                                    const OESystem::OEInterface& itf)
```

Initializes an *OEPrepareDepictionOptions* object from the parameters of a given interface.

See also

- *OEPrepareDepictionOptions* class
- *OEConfigurePrepareDepictionOptions* function

4.3.98 OESetupReportOptions

```
bool OESetupReportOptions(OEReportOptions& opts,
                          const OESystem::OEInterface& itf)
```

Initializes an *OEReportOptions* object from the parameters of a given interface.

See also

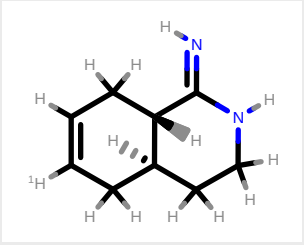
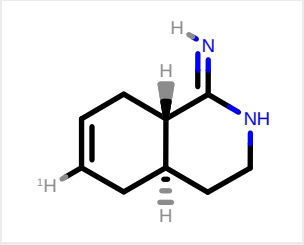
- *OEReportOptions* class
- *OEConfigureReportOptions* function

4.3.99 OESuppressNonDepictionHydrogens

```
bool OESuppressNonDepictionHydrogens(OEChem::OEMolBase& mol)
```

The *OESuppressNonDepictionHydrogens* function suppresses all non-essential hydrogens of the molecule. Only the following hydrogens will be kept:

- hydrogens with (non-zero) isotopic mass
- hydrogens that are necessary to faithfully represent tetrahedral atom stereochemistry or cis/trans bond stereochemistry (defined by the *OEHasDepictionHydrogens* function).

Before	After
	

See also

- *OEHasDepictionHydrogens* function

4.3.100 OEWrite2DRingDictionaryReport

```
bool OEWrite2DRingDictionaryReport(const std::string& filename,
                                   const OEChem::OE2DRingDictionary& ringdict,
                                   const OEReportOptions& opts)
bool OEWrite2DRingDictionaryReport(OEPlatform::oeostream& os,
                                   const std::string& extension,
                                   const OEChem::OE2DRingDictionary& ringdict,
                                   const OEReportOptions& opts)
```

Writes a report of the 2D ring layouts stored in the given ring dictionary. See example in *Table Example of ring dictionary report*.

filename
The filename into which the report is written. The file extension determines the type of the multi-page image.

os
The stream into which the report is written.

ext
The extension which determine the type of the multi-page report.

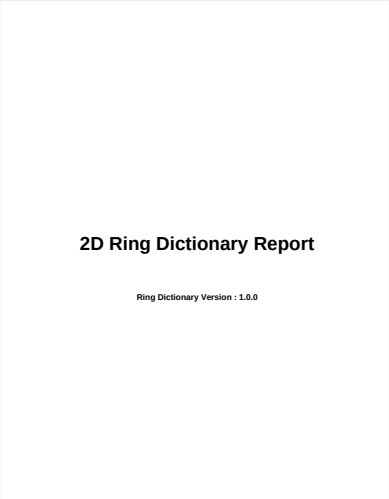
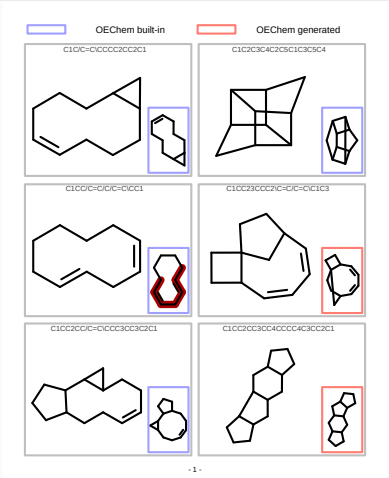
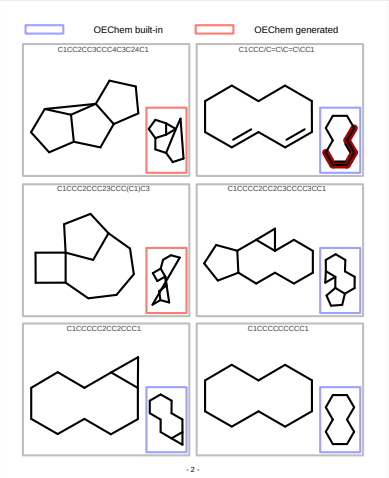
ringdict
The OE2DRingDictionary object that stores ring 2D layouts that can be used to plug into the 2D coordinate generation system.

opts
The *OEReportOptions* object that determines the layout of the pages of the report.

The report depicts each ring systems stored in the ring dictionary.

- The string at the top is the canonical isomeric SMILES that is used as the identifier for the given ring system.
- The larger depiction on the left shows the user-defined layout of the ring system
- The smaller depiction on the right illustrates the default layout of the ring system *i.e.* when the 2D coordinates are generated by **OEChem TK** without utilizing the given user-defined ring dictionary.
  - The blue box indicates that **OEChem TK** has already have a built-in layout for the given ring system.
  - The pink box indicates that the layout of that ring system would be generated on the fly by default without utilizing any built-in templates.
- There are cases when cis/trans bonds are marked on the macro-cycles depicted inside the blue/pink boxes. These annotations indicate potential depiction problem in **OEChem TK**.

Table 55: **Example of ring dictionary report** (The pages are reduced here for visualization convenience)

cover page	page 1	page 2
		

See also

- OE2DRingDictionary class
- *OEAnnotateDepictionProblems* function
- 2D Coordinate Generation Examples section of the **OEChem TK** manual that shows examples about how to generate and utilize user-defined ring dictionaries.

4.3.101 OEWriteImage

Table 56: **The overloaded versions of the OEWriteImage function**

Link	Description
OEWriteImage(filename, image)	writes image into file
OEWriteImage(stream, ext, image)	writes image into output stream

```
bool OEWriteImage(const std::string& filename, const OEImage& image)
```

Writes the image into a file with the given name.

filename
The name of the file into which the image is written. The extension of the filename determine the type of the image.

image
The *OEImage* object which is being written into the given file.

See also

- *OEIsRegisteredImageFile* function

```
bool OEWriteImage(OEPlatform::oeostream& os, const std::string& ext,
                  const OEImage& image)
```

Writes the image into the given output stream.

os
The stream into which the image is written.

ext
The extension which determine the type of the image.

image
The *OEImage* object which is being written into the stream.

See also

- *OEIsRegisteredImageFile* function

4.3.102 OEWriteImageToString

```
std::string OEWriteImageToString(const std::string& ext,
                                 const OEImage& image)
```

Writes the image into a string.

ext
The extension which determine the type of the image.

image
The *OEImage* object which is being written into the string.

Example:

```
OEGraphMol mol = new OEGraphMol();
oechem.OESmilesToMol(mol, "c1ccccc1");
oedepict.OEPrepareDepiction(mol);

OEImage image = new OEImage(200, 200);
oedepict.OERenderMolecule(image, mol);
byte [] bytes = oedepict.OEWriteImageToByteArray("svg", image);
```

See also

- *OEIsRegisteredImageFile* function

4.3.103 OEWriteMultiPageImage

```
bool OEWriteMultiPageImage(const std::string& filename,
                           const OEMultiPageImageFile& image)
```

Writes the multi-page image into a file with the given name.

filename
The name of the file into which the image is written. The extension of the filename determine the type of the image.

image
The *OEMultiPageImageFile* object which is being written into the given file.

See also

- *OEIsRegisteredMultiPageImageFile* function

```
bool OEWriteMultiPageImage(OEPlatform::oeostream& os,
                           const std::string& extension,
                           const OEMultiPageImageFile& image)
```

Writes the multi-page image into the given output stream.

os
The stream into which the multi-page image is written.

ext
The extension which determine the type of the image.

image
The *OEMultiPageImageFile* object which is being written into the stream.

See also

- *OEIsRegisteredMultiPageImageFile* function

4.3.104 OEWriteReport

Table 57: **The overloaded versions of the OEWriteReport function**

Link	Description
OEWriteReport(filename, report)	writes report into file
OEWriteReport(stream, ext, report)	writes report into output stream

```
bool OEWriteReport(const std::string& filename, const OEReport& report,
                   unsigned int page = 0u)
```

Writes page(s) of the report into a file with the given name.

filename
The filename into which the page(s) of the report are written. The file extension determines the type of the multi-page image.

report
The *OEReport* object of which page(s) is/are being written into the given file.

pageidx
This value has to be in the range from *0* to *OEReport.NumPages()*.

Note

If $pageidx$ is a valid page number *i.e* in the range from *1* to *OEReport.NumPages()*, then only the $pageidx^{th}$ page of the report is written into the file. If $pageidx$ is zero and the file extension is recognized as a multi-page format, than all pages of the report are written into the file.

See also

- *OEIsRegisteredImageFile* function
- *OEIsRegisteredMultiPageImageFile* function

```
bool OEWriteReport(OEPlatform::oeostream& os,
                   const std::string& extension, const OEReport& report,
                   unsigned int page = 0u)
```

Writes page(s) of the report into the given stream.

os
The stream into which the page(s) of the report are written.

ext
The extension which determine the type of the image.

report
The *OEReport* object of which page(s) is/are being written into the given file.

pageidx
This value has to be in the range from *0* to *OEReport.NumPages()*.

Note

If $pageidx$ is a valid page number *i.e* in the range from *1* to *OEReport.NumPages()*, then only the $pageidx^{th}$ page of the report is written into the stream. If $pageidx$ is zero and the file extension is recognized as a multi-page format, than all pages of the report are written into the stream.

See also

- *OEIsRegisteredImageFile* function
- *OEIsRegisteredMultiPageImageFile* function

- *OEWriteReportPageByPage* function

4.3.105 OEWriteReportPageByPage

```
bool OEWriteReportPageByPage(const std::string& filename,
                             const OEReport& report)
```

A convenient function that writes pages of the report into the separate numbered files.

filename
The filename that serves as a prefix to write the report page by page. For example, if filename `test.svg` is given, the pages of the report will be written page by page into files: `test-01.svg`, `test-02.svg` ... The extension of file determines the type of the image.

report
The *OEReport* object of which pages are being written into the separate files.

See also

- *OEIsRegisteredImageFile* function
- *OEIsRegisteredMultiPageImageFile* function
- *OEWriteReport* function

4.3.106 OEWriteReportToString

```
std::string OEWriteReportToString(const std::string& ext,
                                  const OEReport& report,
                                  unsigned int page = 0u)
```

Writes the report into a string.

ext
The extension which determine the type of the image.

report
The *OEReport* object of which page(s) is/are being written into the string.

pageidx
This value has to be in the range from *0* to *OEReport.NumPages()*.

Note

If $pageidx$ is a valid page number *i.e* in the range from *1* to *OEReport.NumPages()*, then only the $pageidx^{th}$ page of the report is written into the string. If $pageidx$ is zero and the file extension is recognized as a multi-page format, than all pages of the report are written into the string.

Example:

```
OEGraphMol mol = new OEGraphMol();
oechem.OESmilesToMol(mol, "c1ccccc1");
```

(continues on next page)

(continued from previous page)

```
oedepict.OEPrepareDepiction(mol);

OEReport report = new OEReport(3, 1);
oedepict.OERenderMolecule(report.NewCell(), mol);
byte [] bytes = oedepict.OEWriteReportToByteArray("pdf", report);
```

See also

- *OEIsRegisteredMultiPageImageFile* function

CHAPTER

FIVE

APPENDICES

5.1 Appendix: Element coloring (CPK)

The following two images show the default colors associated with various elements when using the *OEAtomColorStyle.WhiteCPK* style.

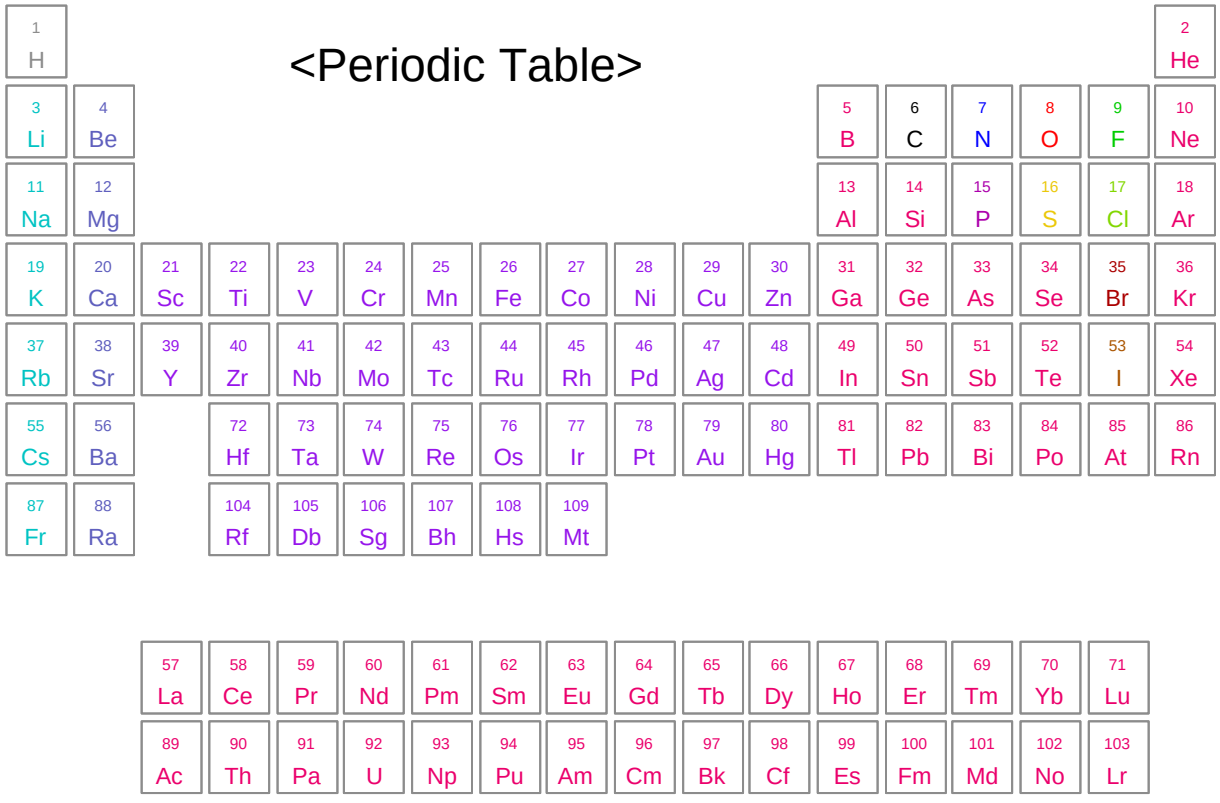


Fig. 1: **Default CPK colors for white background**

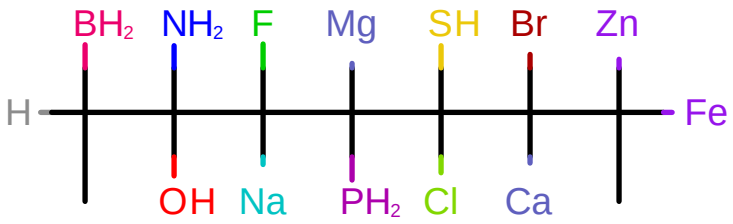


Fig. 2: **Example of molecule depiction with default CPK colors in a white background**

The following two images show the default colors associated with various elements when using the *OEAtomColorStyle.BlackCPK* style.

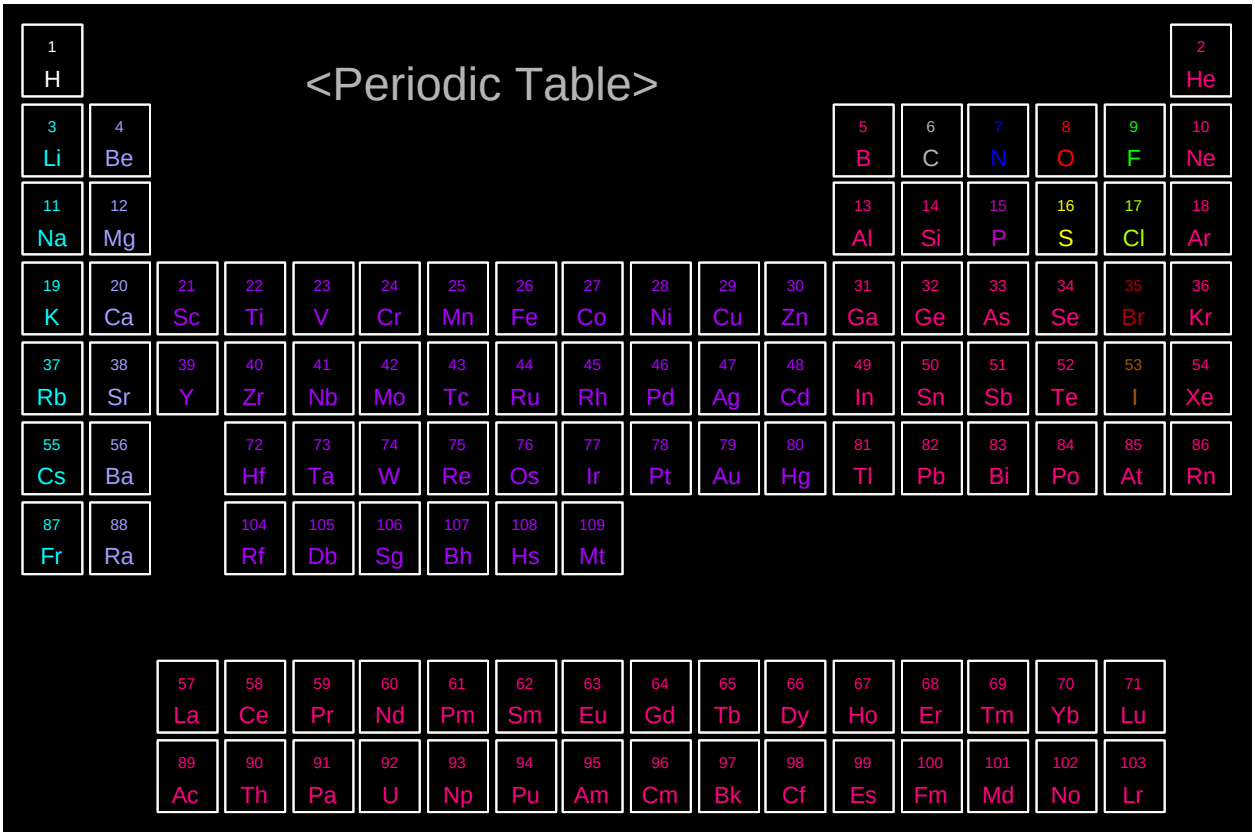


Fig. 3: **Default CPK colors for black background**

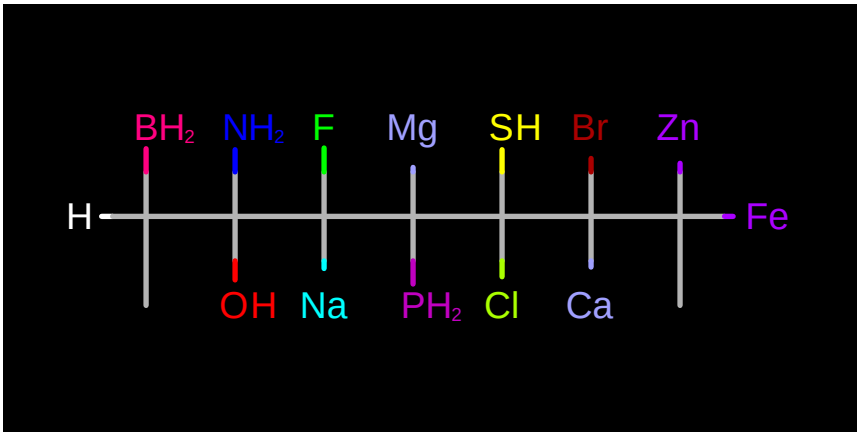


Fig. 4: **Example of molecule depiction with default CPK colors in a black background**

CHAPTER

SIX

RELEASE HISTORY

6.1 OEDepict TK 2.7.0

6.1.1 New Features

- Two new methods, *GetOptimizeMacrocycles* and *SetOptimizeMacrocycles*, have been added to *OEPrepareDepictionOptions* to enable enhanced algorithmic depiction of macrocycle rings.

6.2 OEDepict TK 2.6.0

Minor internal improvements have been made.

6.3 OEDepict TK 2.5.6

Minor internal improvements have been made.

6.4 OEDepict TK 2.5.5

Minor internal improvements have been made.

6.5 OEDepict TK 2.5.4

Minor internal improvements have been made.

6.6 OEDepict TK 2.5.3

Minor internal improvements have been made.

6.7 OEDepict TK 2.5.2

Minor internal improvements have been made.

6.8 OEDepict TK 2.5.1

Minor internal improvements have been made.

6.9 OEDepict TK 2.5.0

- Minor internal improvements have been made.

6.10 OEDepict TK 2.4.7

Fall 2021

- Minor internal improvements have been made.

6.11 OEDepict TK 2.4.6

July 2021

Minor internal improvements have been made.

6.12 OEDepict TK 2.4.5

Fall 2020

6.12.1 New features

- A new function, *OEGetLabelPosition*, that identifies the position where atoms labels are added.
- Approximately 40 new protective groups have been added to the namespace *OEProtectiveGroupStyle*.
- A new function, *HasProtectiveGroupStyle*, has been added that determines whether a given protective group style has been set in *OE2DMolDisplayOptions*.
- The function *GetProtectiveGroupStyle* has been deprecated. It can no longer be used to determine if multiple protective group styles from *OEProtectiveGroupStyle* have been set. *HasProtectiveGroupStyle* can now be used instead.

6.12.2 Minor bug fixes

- An issue that caused *OEPrepareAlignedDepiction* to generate incorrect stereo depictions when *SetClearCoords* was set to `false` has been fixed.
- Figures have been added to show depiction with and without the new protective groups to *OEProtectiveGroupStyle*.

6.13 OEDepict TK 2.4.4

6.13.1 New features

- Two new methods, *OE2DMolDisplayOptions.SetExplicitAtomLabelAngle* and *OE2DMolDisplayOptions.GetExplicitAtomLabelAngle*, have been added that control when explicit "C" labels will be displayed on the molecule graph.

6.14 OEDepict TK 2.4.3

6.14.1 Minor bug fixes

- The signature of the *OECreateWin32GraphicsImage* function has been changed to use std::shared_ptr.

- SVG group IDs and names are now limited to 64 characters.

6.14.2 Python-specific changes

- User-defined atom and bond SVG markings are now supported in Python. See the examples in the *OEAtomSVGMarkupBase* and *OEBondSVGMarkupBase* base classes.

6.15 OEDepict TK 2.4.2

6.15.1 New features

- An option has been added to the molecule alignment process that allows it to accept alignments with a limited number of atom pair clashes in the fixed coordinates mode. See also the *OEAlignmentOptions.GetMaxAllowedAtomPairClashes* and *OEAlignmentOptions.SetMaxAllowedAtomPairClashes* methods of the *OEAlignmentOptions* class.
- In fixed coordinate alignment mode (see the *OEAlignmentOptions.SetFixedCoords* method), the order in which single bonds are rotated in order to eliminate atom clashes has been improved.
- A new constructor has been added that allows the *OEAlignmentResult* class to be initialized from an OEMatchBase object.

6.15.2 Major bug fixes

- A bug in the alignment process that occurred when the number of rotatable bonds of the fit molecule exceeded the built-in limit has been fixed.

6.15.3 Minor bug fixes

- A minor memory leak in the molecule alignment process has been fixed.
- On macOS, the deprecated `kCTLeftTextAlignment`, `kCTRightTextAlignment`, and `kCTCenterTextAlignment` Core Graphics constants have been replaced with new APIs. These constants are used to handle text alignment when drawing in PNG images.

6.15.4 Documentation changes

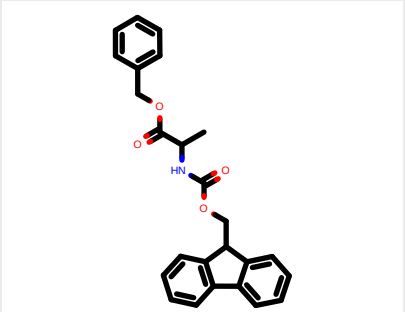
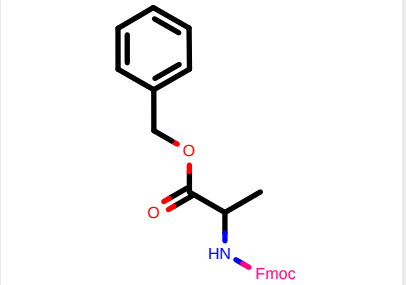
- The example in the *Depicting MDL Query Substructure Search Hits* section has been revised.

6.16 OEDepict TK 2.4.1

6.16.1 New features

- The following APIs have been added to generate consistent legend layouts:
  - *OELegendLayoutStyle* namespace
  - *OELegendColorStyle* namespace
  - *OELegendInteractiveStyle* namespace
  - *OELegendLayoutOptions* class
  - *OELegendLayout* class
  - *OEDrawLegendLayout* function
- The following APIs have been added to depict common protective groups:

  - *OEProtectiveGroupStyle* namespace
  - *OE2DMolDisplayOptions.SetProtectiveGroupLabelFont*
  - *OE2DMolDisplayOptions.GetProtectiveGroupLabelFont*
  - *OE2DMolDisplayOptions.SetProtectiveGroupStyle*
  - *OE2DMolDisplayOptions.GetProtectiveGroupStyle*

default depiction	using *OEProtectiveGroupStyle.Fmoc* style
	

- The following APIs have been added to mark up the bonds in SVG images generated by the *OERenderMolecule* functions:
  - *OE2DMolDisplayOptions.SetBondSVGMarkupFunctor*
  - *OE2DMolDisplayOptions.GetAtomSVGMarkupFunctor*
  - *OEBondSVGMarkupBase*
  - *OEBondSVGNoMarkup*
  - *OEBondSVGBondIdxMarkup*
- A new *OEDrawHighlighting* function has been added to render molecule highlights into a separate image rather than into the molecule display.
- New *OEAddHighlighting* function overloads have been added that take an OESubSearch object and highlight its matches on the molecule display.
- New *OEAddWatermark* functions have been added that render a light gray watermark onto an image or a molecule display.
- A new *OEImageBase.GetSVGGroups* method has been added to return the SVG groups associated with the image.
- New *OERotateVector* and *OELengthenVector* functions have been added.
- A new *OEAddSVGHover* function has been added that takes two target SVG groups to create a "hover" effect.

6.16.2 Minor bug fixes

- The minimum margin allowed for *OE2DMolDisplayOptions* has been reduced from `5.0%` to `2.5%`.
- A duplicate SVG group warning is no longer thrown when transferring built-in SVG groups from one image to another.

6.17 OEDepict TK 2.4.0

6.17.1 New features

- The following API for marking up SVG images, introduced in the 2017.Jun and 2017.Oct releases, is no longer preliminary. It is now also available in both Java and C#:
  - *OESVGClass* class
  - *OESVGGroup* class
  - *OEAtomSVGAtomIdxMarkup* class
  - *OEAtomSVGMarkupBase* class
  - *OEAtomSVGNoMarkup* class
  - *OEAtomSVGResidueMarkup* class
  - `OEAddBuiltInHiddenSVGClass` function
  - `OEAddBuiltInVisibleSVGClass` function
  - `OEAddSVGClickEvent` function
  - `OEDrawSVGHoverText` function
  - `OEDrawSVGToggleText` function
  - `OEGetBuiltInHiddenSVGClass` function
  - `OEGetBuiltInVisibleSVGClass` function
- Two new functions, `OEIsValidSVGGroupId` and `OEIsValidSVGClassName`, have been added.
- Two new low-level functions, `OEAddSVGHover` and `OEAddSVGToggle`, have been added to generate hover and toggle effects, respectively, between two SVG groups.
- The *OEAtomSVGResidueMarkup* class now takes a separator option.
- A new function, `OEAddLabel`, has been added to render a label on an image at a user-defined location.
- Two new functions, `OEAddOpenEyeIcon` and `OEAddInteractiveIcon`, and the namespace `OEIconLocation` have been added.
- Two new methods, `OE2DPath.GetStart` and `OE2DPath.GetEnd`, have been added.
- Two new pens, `OETransparentPen` and `OESVGAreaPen`, have been added.

6.17.2 Java-specific changes

- The API for marking up SVG images is now fully supported in Java (see the list above).

6.17.3 C#-specific changes

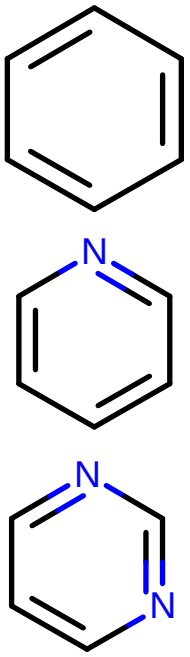
- The API for marking up SVG images is now fully supported in C# (see the list above).

6.18 OEDepict TK 2.3.6

6.18.1 New features

- The following API has been added to increase the reporting capabilities of **OEDepict TK**:
  - *OEImageTable* class

- *OEImageTableOptions* class
- `OEImageTableStyle` namespace



Property	Value
Name	benzene
SMILES	c1ccccc1
MV	78.112

Property	Value
Name	pyridine
SMILES	c1ccncc1
MV	79.100

Property	Value
Name	pyrimidine
SMILES	c1cncnc1
MV	80.088

Fig. 1: **Example of depicting molecules and data using image table**

- The following API has been added to render labels associated with certain parts of the depicted molecule:
  - *OEHighlightLabel* class
  - `OEAddLabel` function

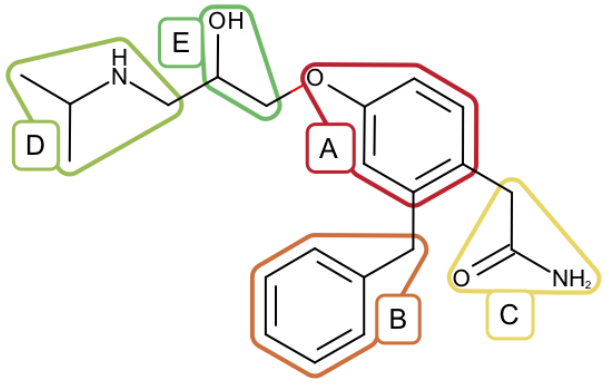


Fig. 2: **Example of adding labels on molecule depiction**

- A new function, `OEDrawText`, has been added.
- The constructor of the `OEHighlightByLasso` class now takes a parameter that determines the width of the "lasso" used for highlighting. The corresponding `GetLineWidthScale` and `SetLineWidthScale` methods

have been added.

- A new constructor has been added to the *OEImageFrame* class that generates a new image with equal padding around it.
- A new function, *OEWriteReportPageByPage*, has been added.
- A new constructor has been added to the *OE2DPath* class that copies and rotates an already existing path.

6.18.2 Preliminary API - New Functionalities

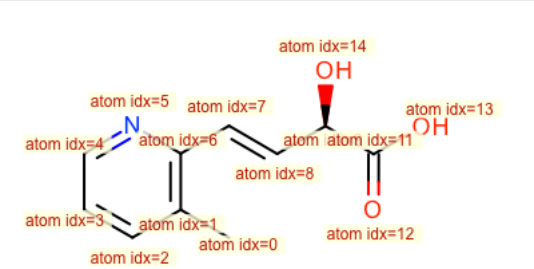
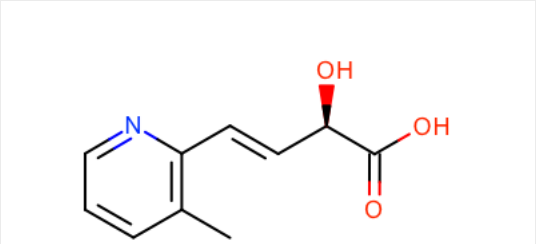
- A new overload of the function *OEAddSVGClickEvent* has been added that allows associating any *OESVGGroup* object with an SVG click event.
- A new Preliminary API has been added to generate and access built-in visibility SVG classes (*OESVGClass*):
  - *OEAddBuiltInHiddenSVGClass*
  - *OEGetBuiltInHiddenSVGClass*
  - *OEAddBuiltInVisibleSVGClass*
  - *OEGetBuiltInVisibleSVGClass*
- A new Preliminary API has been added to mark up the atoms in SVG images generated by the *OERenderMolecule* function:
  - *OE2DMolDisplayOptions.SetAtomSVGMarkupFunctor*
  - *OE2DMolDisplayOptions.GetAtomSVGMarkupFunctor*
  - *OEAtomSVGMarkupBase*
  - *OEAtomSVGNoMarkup*
  - *OEAtomSVGAtomIdxMarkup*
  - *OEAtomSVGResidueMarkup*
- A new constant, *OELayerPosition.SVG*, has been added to the *OELayerPosition* namespace.
- The following virtual methods have been added to the *OEImageFileBase* class:
  - *OEImageFileBase.PushGroup*
  - *OEImageFileBase.PopGroup*
  - *OEImageFileBase.IsVisible*
- A new method, *OESVGGroup.IsVisible*, has been added.

6.18.3 Preliminary API - API Changes

- The atom and bond display parameters of the following functions are no longer `const`:
  - *OEDrawSVGToggleText*
  - *OEDrawSVGHoverText*
  - *OEAddSVGClickEvent*

6.18.4 Major bug fixes

- Objects that have been drawn in hidden SVG groups are no longer visible in other image file formats. For example, when adding hover text to atoms (using the *OEDrawSVGHoverText* function), the text will only be visible in SVG images when the mouse is moved over a displayed atom. If the image is written into any other image file format, the hover text is no longer visible.

2017.Jun	2017.Oct
	

- When generating interactive SVG images, hover text (*OEDrawSVGToggleText*) and click events (*OEAddSVGClickEvent*) can both be added to atoms and bonds.

6.18.5 Minor bug fixes

- The Ogham API, deprecated since 2014, is no longer available.

6.18.6 Documentation changes

- A new section, *Depicting Molecules in a Table*, has been added.

6.19 OEDepict TK 2.3.5

6.19.1 New features

- A new function, *OEConfigure2DMolDisplayOptions*, has been added to help generate consistent command line interfaces for depiction examples and utilities. Generated interfaces now include correspondence between each command line option and its related entrypoint in **OEDepict TK**. For more details see the documentation for *OEConfigure2DMolDisplayOptions*.
- A position property has been added to the "lasso" highlighting style that allows drawing the highlights either below or above the depicted molecules. This property can be set either by the constructor of the *OEHighlightByLasso* class or via the *OEHighlightByLasso.SetPosition* method.
- The *OE2DPath* class has been extended to handle "empty" 2D paths:
  - *OE2DPath* default constructor
  - *OE2DPath.AddStartPoint* method
  - *OE2DPath.SetClosed* method
  - *OE2DPath.IsValid* method

6.19.2 Preliminary API

- A new Preliminary API has been added to support the generation of interactive SVG images:
  - *OESVGClass* and *OESVGGroup* classes
  - *OEImageBase.PushGroup*, *OEImageBase.PopGroup*, *OEImageBase.NewSVGGroup*, *OEImageBase.NewSVGClass*, *OEImageBase.GetSVGGroup*, and *OEImageBase.GetSVGClass* methods of the *OEImageBase* base class and its derived classes
  - *OEDrawSVGHoverText*, *OEDrawSVGToggleText*, and *OEAddSVGClickEvent* functions

6.19.3 Major bug fixes

- On OSX, the deprecated `CGContextSelectFont` and `CGContextShowTextAtPoint` Core Graphics functions have been replaced. These function were used to handle fonts when drawing text in PNG images. **OEDepict TK** currently supports the following font mapping on OSX for PNG images:

OpenEye Font Family Type	Font Name
OEFontFamily.Arial	`Arial`
OEFontFamily.Courier	`CourierNewPS`
OEFontFamily.Helvetica	`Helvetica`
OEFontFamily.Times	`TimesNewRomanPS`

6.19.4 Minor bug fixes

- When performing fixed coordinate molecule alignment (see *OEAlignmentOptions.SetFixedCoords*), only the set of atoms considered during the alignment are now used to evaluate the quality of the alignment. As a result, fixed coordinate alignment succeeds more frequently, circumventing the need to fall back to the more exhaustive alignment method.
- Bond stereo information is now re-perceived by calling the `OEMDLPerceiveBondStereo` function after finding the best 2D alignment with the *OEPrepareAlignedDepiction* function when the alignment is defined by either an OESubSearch or an OEMCSSearch object.

6.19.5 Python-specific changes

- The following Python functions have been added:
  - `OERenderMoleculeToBytes`
  - `OEWriteImageToBytes`
  - `OEWriteReportToBytes`

6.19.6 Documentation changes

- A new chapter, *Generating Interactive SVG Images*, has been added.
- A new section, *Python Cookbook Examples*, has been added.

6.20 OEDepict TK 2.3.4

6.20.1 New features

- A new function, *OEPrepareMultiAlignedDepiction*, has been added that performs 2D molecule alignment based on common fragments between two molecules.

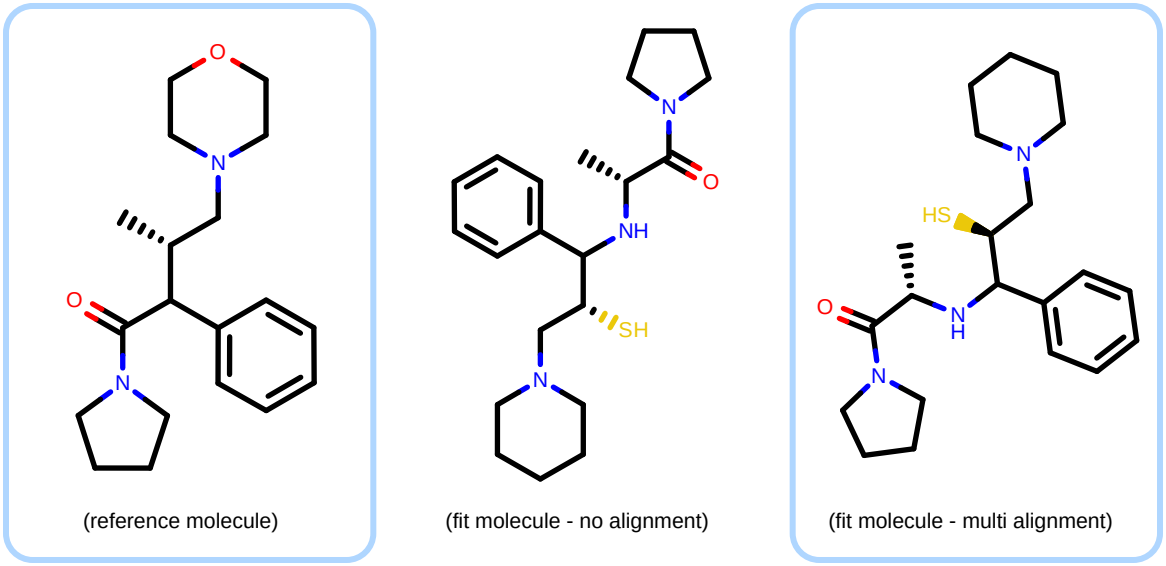


Fig. 3: **Example of depiction with alignment based on molecule similarity**

- A new method, *CreateCopy*, has been added to the *OEHighlightBase* base class and all of its derived classes.
- A new function, *OEDrawCurvedRectangle*, has been added that renders a rectangle with rounded corners.

6.20.2 Major bug fixes

- The *OE2DMolDisplay* constructor previously threw an error when initialized with an empty molecule or with a molecule with no coordinates. A warning is now thrown and the object is marked invalid (e.g., the *OE2DMolDisplay.IsValid* method will return `false`). In addition, an *OE2DMolDisplay* object becomes invalid if the molecule from which it was initialized is modified (e.g., when atoms/bonds are added or deleted). All **OEDepict TK** functions handling an *OE2DMolDisplay* object now check the validity of the object prior to use.
- Reactions with only product or only reactant components are now depicted correctly with a reaction arrow and the "+" sign between the components.
- The methods *OEImageBase.DrawArc* and *OEImageBase.DrawPie* have been unified when generating PDF images. As a result, arc and pie objects in PDF images appear more seamless.

6.20.3 Minor bug fixes

- When generating an SVG image, the opacity of the background is now written into the file. This allows generating images with a semi-transparent background.
- For the molecules C#C and C=C, the carbon atom labels are now always explicitly depicted.
- When determining the scaling of a depicted molecule, the bounding box of each multi-line bond is now considered to ensure that no bond is rendered outside the canvas.
- A minor molecule scaling issue that occurred when using fixed atom label fonts has been fixed.
- The function *OEDrawChevronArrow* has been improved to handle very short arrows. In addition, the default parameters of the function have been changed.

6.20.4 Documentation changes

- A new section, *Molecule Alignment Based on Molecular Similarity*, has been added to demonstrate the new *OEPrepareMultiAlignedDepiction* function.
- Warnings have been added to *OEWriteReportToString*, *OERenderMoleculeToString*, and *OEWriteImageToString* to alert that these functions return bytes in Python 3.

6.21 OEDepict TK 2.3.3

6.21.1 New features

- The following APIs have been added to allow the interrogation of the *OE2DPath* class:
  - *OE2DPath.GetPoints* and *OE2DPath.IsClosed* methods
  - *OE2DPathPoint* class
  - *OE2DPathPointType* namespace

6.21.2 Minor bug fixes

- *OEPrepareDepiction* has been fixed to correctly suppress all hydrogens when requested. Previously, it retained depiction hydrogens for some molecules.
- *OEPrepareDepiction* no longer crashes when used on `OEMolBaseType.OEMiniMol` molecule implementations.

6.21.3 Documentation changes

- All images in the **OEDepict TK** documentation have been regenerated to reflect the changes made since the 2016.Jun release.

6.22 OEDepict TK 2.3.2

6.22.1 New features

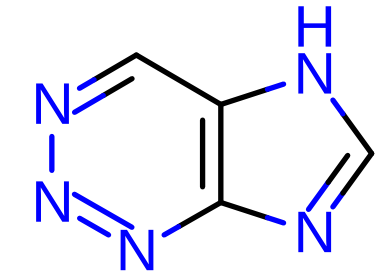
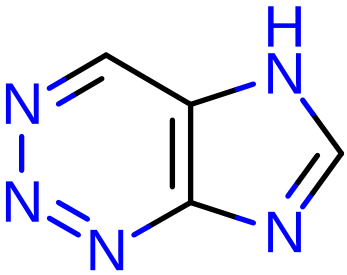
- *OE2DMolDisplayOptions.GetAtomVisibilityFunctor* and *OE2DMolDisplayOptions.SetAtomVisibilityFunctor* methods have been added to support the ability to selectively hide portions of a structure from depictions.
- *OEAlignmentOptions.SetAddDepictionHydrogens* and *OEAlignmentOptions.GetAddDepictionHydrogens* methods have been added to control depiction hydrogens added during molecule alignment.
- *OEGetAverageBondLengthInDispCoords* function has been added.
- *OEGetNearbyAtom* and *OEGetNearbyBond* functions have been added.
- *OENearestAtom* and *OENearestBond* classes now store the display position of the nearest atom and bond, respectively.

6.22.2 Major bug fixes

- A memory issue that caused *OEPrepareAlignedDepiction* function to crash has been fixed. The performance of molecule alignment has been also improved.

6.22.3 Minor bug fixes

- Clashes between atom labels and inside bond lines of rings have been eliminated in cases when font scales larger than the default value were used. See the example below.

2016.Feb	2016.Jun
	

6.22.4 C#-specific changes

OEImageBase.DrawArc method no longer throws an "Out Of Memory" error message when drawing an arc with a small angle and/or small radius. For more details, see Graphics.DrawArc Method.

6.22.5 Documentation changes

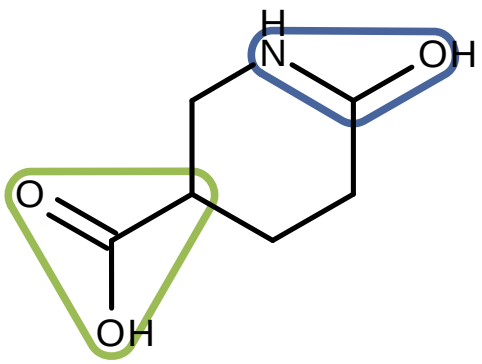
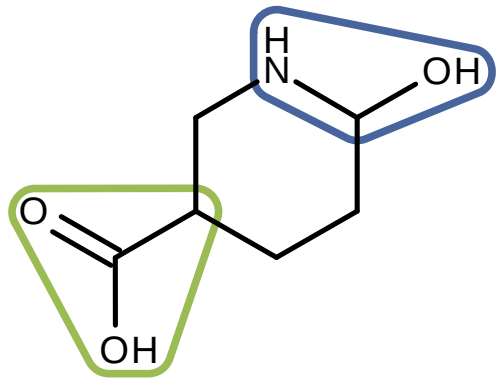
- All images in the **OEDepict TK** documentation were automatically regenerated to reflect the changes made since the previous release.

6.23 OEDepict TK 2.3.1

6.23.1 New features

- *OEHighlightByLasso.SetConsiderAtomLabelBoundingBox* method has been added to keep lines of highlighting from clashing with the atom labels.

Table 1: **Examples of highlighting using the lasso style**

default style	new style
	

6.23.2 Minor bug fixes

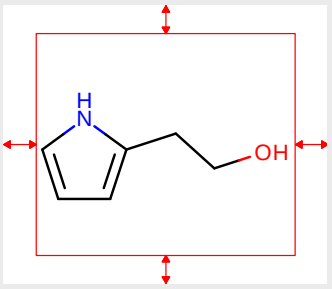
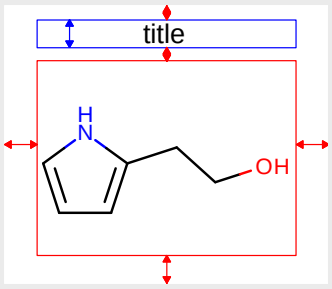
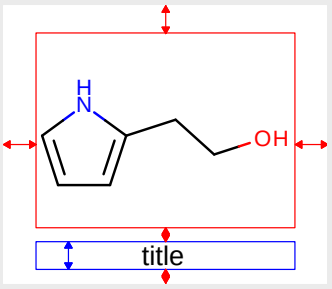
- Bonds with the `OEBondStereo.DoubleEither` property are now always depicted with the *`OEBondDisplayType.DoubleBowTie`* style.

6.24 OEDepict TK 2.3.0

6.24.1 New features

- Functionalities have been added to customize the general layout of the molecule display:
  - the margin(s) around the molecule can now be changed (for more details, see *`OE2DMolDisplayOptions.SetMargin`* and *`OE2DMolDisplayOptions.SetMargins`* methods)
  - the height of the title area can now be changed (for more details, see *`OE2DMolDisplayOptions.SetTitleHeight`* method)

Table 2: **Example of molecule display layouts depending on position of the title**

title hidden	title at the top	title at the bottom
		

- A new highlighting style, *`OEHighlightStyle.Lasso`*, has been added along with the corresponding highlighting class *OEHighlightByLasso*. See the example in *Figure: Example of highlighting using the "lasso" style*. below.

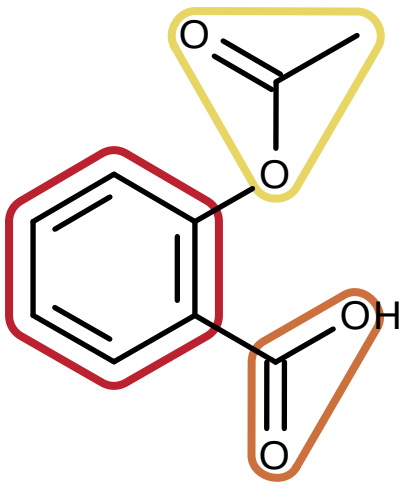
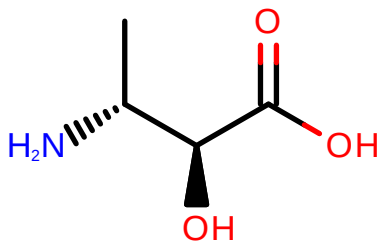
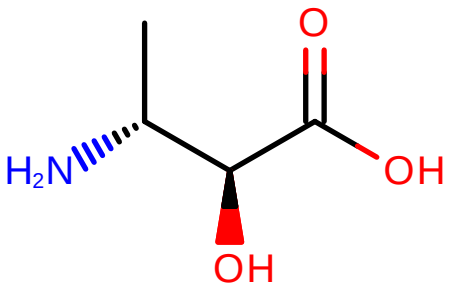


Fig. 4: **Example of highlighting using the "lasso" style**

- Text can now be associated with hyperlinks via the *OEFont* class. This functionality is only available for PDF and SVG image formats. For more details and an example see the *OEFont.SetHyperlink* method.
- The title of the molecule is now scaled along with the size of the image by default. The *OE2DMolDisplayOptions.SetTitleFontScale* method was added to enable changing the scaling factor or allowing a fixed font size to be used regardless of the dimensions of the image. See also the *OE2DMolDisplayOptions.SetTitleFont* method.
- The *OE2DMolDisplayOptions.SetAtomLabelFontScale* method now allows using a fixed font size regardless of size of the molecule. Previously, the atom labels of the molecule were scaled along with the size of molecule.
- In case of the *OEBondColorStyle.ByElement* style, the wedge/hash stereo bonds are also colored by the corresponding element types. See the effect of this change in *Table: Example of bond coloring* below.

Table 3: **Example of bond coloring**

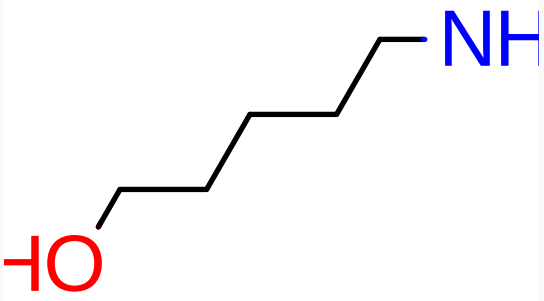
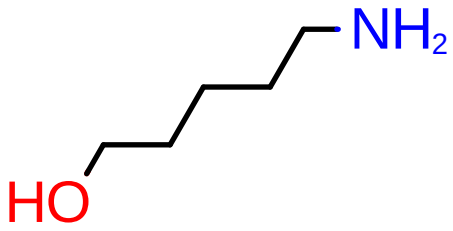
OEDepict TK 2.2.6 (2015.Jun)	OEDepict TK 2.3.0 (2015.Oct)
	

- If the width or the height of a molecule display is not specified it will be set to `50.0`.
- *OEWriteReportToString* function has been added.
- *OEDrawCurvedBorder* function has been added.

6.24.2 Major bug fixes

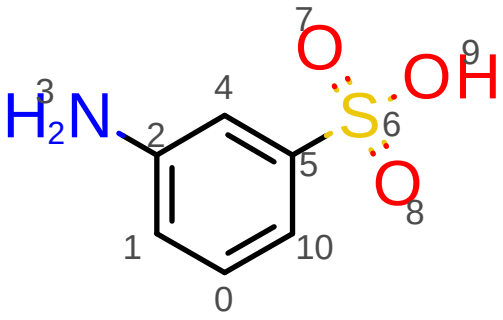
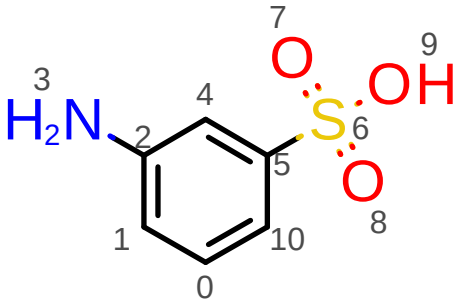
- The layout of the molecule depiction has been improved to eliminate atom label clippings. These label clippings used to occur in cases when font scales larger than the default value were used. See the effect of this change in *Table: Improved depiction layout* below.

Table 4: **Improved depiction layout**

OEDepict TK 2.2.6 (2015.Jun)	OEDepict TK 2.3.0 (2015.Oct)
	

- The bounding boxes of labels are now considered when positioning atom properties. As a result, the number of overlaps is reduced in cases where font scales larger than the default value are used. See the effect of this change in *Table: Example of improving the positioning of atom properties* below.

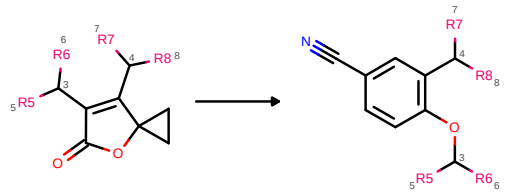
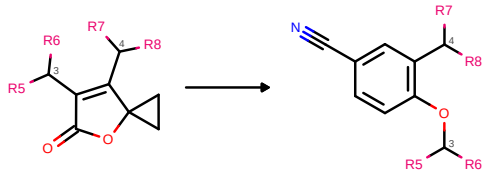
Table 5: **Example of improving the positioning of atom properties**

OEDepict TK 2.2.6 (2015.Jun)	OEDepict TK 2.3.0 (2015.Oct)
	

6.24.3 Minor bug fixes

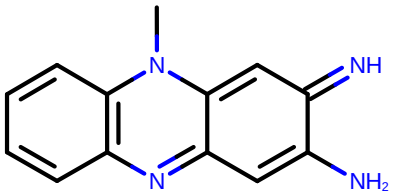
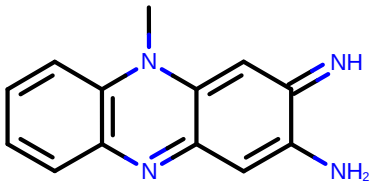
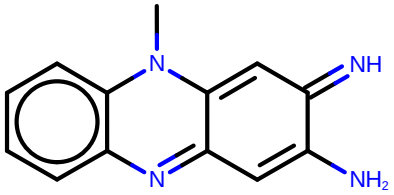
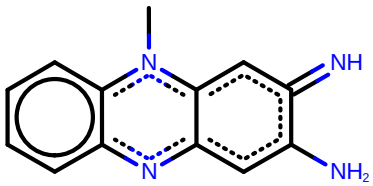
- The default parameter for the maximum number of bond rotations of the *OEAlignmentOptions* class is $2^{16} = 65536$. This means that by default only the first 16 single bonds of the *fit* molecule will be rotated to maximize molecule alignment when invoking the `OEPrepareAlignedDepiction` function.
- `OEPrepareAlignedDepiction` has been optimized to be 10% faster on average. The algorithm was also revised to reduce the accumulation of floating point errors.
- The atom mapping index is no longer displayed (redundantly) in the case of R-groups. See the effect of this change in *Table: Example of displaying atom map indices* below. See also the *OEDisplayAtomMapIdx* class.

Table 6: **Example of displaying atom map indices**

OEDepict TK 2.2.6 (2015.Jun)	OEDepict TK 2.3.0 (2015.Oct)
	

- `OEAromaticStyle.Circle` depiction style has been improved. See the effect of this change in *Table: Example of Kekule (default) and circle aromatic depiction styles* below.

Table 7: **Example of Kekule (default) and circle aromatic depiction styles**

aromaticity style	OEDepict TK 2.2.6 (2015.Jun)	OEDepict TK 2.3.0 (2015.Oct)
`OEAromaticStyle.Kekule`		
`OEAromaticStyle.Circle`		

- If a bond display style cannot be determined - for example, in case of a broken molecule - the bond is now displayed using the `OEBondDisplayType.Any` style and no error message will be thrown.
- `OEAddHighlightOverlay` function no longer generates different images on different platforms.

6.24.4 Python-specific changes

- The deprecated Ogham API is no longer tested in Python. See also *Deprecated Ogham* section.

6.24.5 Documentation changes

- *Molecule Depiction with Transparent Background* section was added to show how to generate molecule depiction with a transparent background.
- Code snippets have been added to the following functions:
  - *OERenderMoleculeToString*
  - *OEWriteImageToString*
- All images in this documentation have been automatically regenerated to reflect the changes made since the previous release.

6.25 OEDepict TK 2.2.6

6.25.1 New features

- The following changes have been made to improve the SVG images containing certain geometries (especially those made by the OE2DPropMap class):
  - *OELineCap.Butt* (the default line cup) and *OELineJoin.Miter* (the default line join) parameters for shapes (lines, rectangles, etc.) are no longer written into the SVG images.
  - *OEImageBase.DrawRectangle* for SVG will now trim the trailing zeros and the decimal point.

  The above changes resulted in 40% smaller images and 10% faster image writing without any visual difference.
- *OE2DPath* class has been added to allow drawing more complex lines connected by a series of points. The following methods were added to accept *OE2DPath* objects:
  - *OEImageBase.DrawPath*
  - *OEImage.DrawPath*
  - *OEImageFrame.DrawPath*
- *OEHighlightByColor.SetAtomExternalHighlight* was added to generate better highlighting for small images.
- *OEAnnotateValenceProblems* function was added to highlight potential atom valence problems on the molecule diagram.
- *OEWrite2DRingDictionaryReport* function was added to generate a multi-page report for user-defined ring dictionaries. See also the 2D Coordinate Generation chapter in the **OEChem TK** manual.
- *OEImageBase.GetGlobalOffset* method was added to return the global position of the image. This is currently only useful for subframes of an *OEImageFrame* object.

6.25.2 Major bug fixes

- *OEPrepareAlignedDepiction* has been optimized to be twice as fast on average.

6.25.3 Minor bug fixes

- *OEAnnotateDepictionProblems* style has been redesigned to be more visually pleasing.
- The positioning of + signs for reactions has been improved. The + signs and the arrow of the reactions now scale with the default line width of the molecule.

6.25.4 Documentation changes

- *ringdict2pdf* example has been added to generate a multi-page ring dictionary report.
- `-ringdict` parameter has been added to the following examples to utilize user-defined ring dictionaries:
  - *depict* example
  - *mol2img* example
  - *mols2pdf* example
- All images in this documentation were automatically regenerated to reflect the changes made since the previous release.

6.26 OEDepict TK 2.2.5

6.26.1 New features

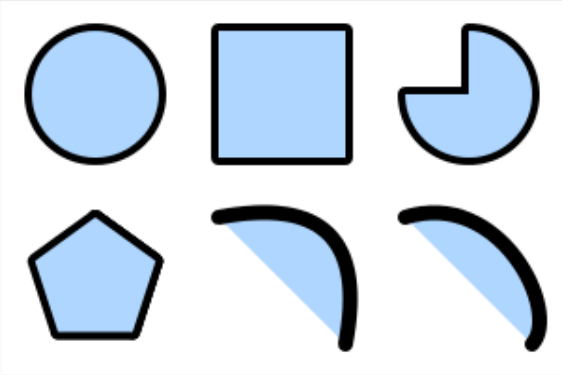
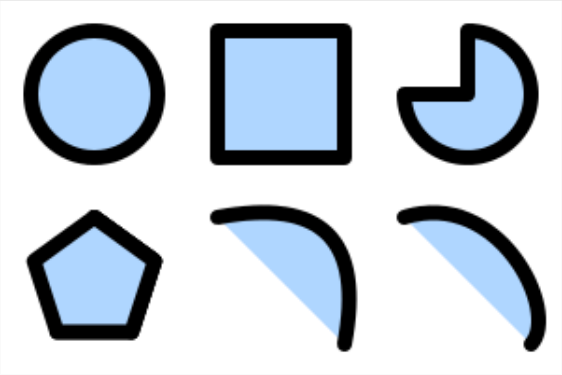
- *OEImage* allows users to easily rescale images without having to re-render all the drawing commands.
- *OEPrepareDepictionOptions.SetDepictOrientation* allows users to control the preferred orientation of the 2D coordinates. This allows optimizing 2D coordinate generation for a particular orientation: *horizontal*, *vertical*, or *square*. For example, *OEDepictOrientation.Horizontal* should be preferred when vertical space is expensive, as on web pages and spreadsheets.
- *OEHighlightByStick.SetAtomExternalHighlightRatio* has been added to generate better highlighting for very small images. See the image in the *OEHighlightByStick.SetAtomExternalHighlightRatio* documentation for an example.
- *OEConfigurePrepareDepictionOptions* provides a convenient way to translate command line parameters into options on a *OEPrepareDepictionOptions* object. All **OEDepict TK** examples have been updated to use these functions for building consistent command line interfaces. See the examples in the *OEDepict Examples Summary* section.
- *OEGetCenter* and *OEDrawTextToCenter* are convenience functions added to ensure that text is drawn vertically and horizontally in the center of an *OEImageBase*.
- Two new pens, *OELightGreyBoxPen* and *OELightGreyPen*, have been added.

6.26.2 Major bug fixes

- *OEPrepareDepiction* now automatically regenerates 2D coordinates when the current 2D coordinates are not in the XY plane. *OEPrepareDepiction* now clears coordinates when any of the following conditions are `true`:
  - *OEPrepareDepictionOptions.GetClearCoords* returns `true`.
  - `OEMolBase.GetDimension` does not return `2`.
  - Any Z coordinate is not `0.0`.

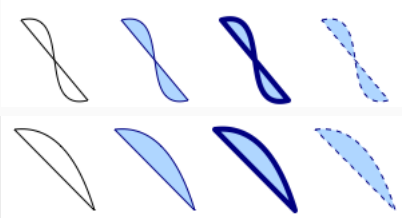
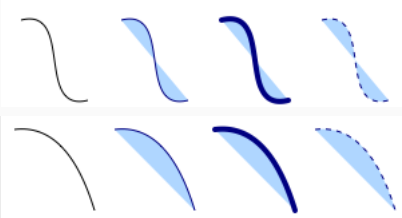
  This protects against corrupted coordinate states that may be caused by file formats not normally used for depiction, such as PDB.
- PNG images generated on Linux will now have properly filled objects and will no longer look different than those generated on OSX or Windows, as shown in the following images:

Table 8: **Example of drawing filled objects with Cairo**

OEDepict TK 2.2.4	OEDepict TK 2.2.5
	

- The cubic and quadratic curves drawn on OSX PNG images are no longer closed and are now consistent with PNG images generated on other platforms.

Table 9: **Example of drawing cubic and quadratic curves on OSX**

OEDepict TK 2.2.4	OEDepict TK 2.2.5
	

- PDF files should no longer cause the following warning message when read into Adobe Acrobat:

```
The font "Times" contains a bad/BBox.
```

6.26.3 Minor bug fixes

- A new *OE2DMolDisplay.IsValid* method has been added. A molecule display is considered valid only if it was initialized with a non-empty molecule with either 2D or 3D coordinates. An invalid molecule display cannot be rendered by the *OERenderMolecule* or the *OERenderMoleculeToString* functions.
- The *OEHydrogenStyle.Hidden* style will no longer show any hydrogen labels, even if the hydrogen has a charge, isotopic mass, or is a radical.
- The *OEAnnotateDepictionProblems* function now displays `E/Z` CIP bond labels rather than `C/T` in cases where double bonds are not correctly depicted according to their cis/trans stereo specifications.

6.26.4 Documentation changes

- All images in this documentation were automatically regenerated to reflect the changes made since the previous release.
- The documentation of all deprecated Ogham classes, functions, and namespaces has been removed:

  Deprecated Ogham classes:

  - OE8BitImage
  - OEAtomStyle
  - OEBondStyle
  - OEDepictBase
  - OEDepictColor
  - OEDepictView
  - OEPSImage
  - OEPoly
  - OEPolyF
  - OESVGImage
  - OEVert
  - OEVertF

  Deprecated Ogham functions:

  - OEWriteBMP
  - OEWriteEPS
  - OEWriteGIF
  - OEWritePPM
  - OEWriteRGB
  - OEWriteXPM

  Deprecated Ogham namespace:

  - OENamedColor

- Minor changes have been made to the documentation and examples for consistency and/or spelling errors.
- More C# examples have been added.

6.26.5 C++-specific changes

- OECreateWin32GraphicsImage now allows interoperability with the Windows GDI+ Graphics class object. This function is only available on the Windows operating system through C++.

6.26.6 Python-specific changes

- **OEDepict TK** no longer provides Python wrapping for the following methods of the deprecated OEDepictView class:
  - GetViewCoords
  - GetForeColor and SetForeColor
  - GetBackColor and SetBackColor

6.27 OEDepict TK 2.2.4

6.27.1 New features

- *OEAnnotateDepictionProblems* added to mark atom and bond stereo coordinate generation problems on output images.
- Added *OEDrawChevronArrow* function.
- Added *OESuppressNonDepictionHydrogens* function.

6.27.2 Major bug fixes

- The *OEPrepareDepiction* function now throws a warning in cases when 2D coordinates are generated that are inconsistent with the cis/trans bond stereo configuration of the molecule.
- *OEPrepareAlignedDepiction* can now handle molecules with more than 32 rotatable bonds. Previously, if the number of rotatable bonds was greater than 32, not enough sampling would be done to achieve a good alignment. Additionally, the set of bonds to rotate is decided in a dynamic fashion to mitigate the expense of fitting large flexible molecules.

6.27.3 Minor bug fixes

- Added the metal colors introduced in the 2014.Jun release to the parameters generated by the *OEConfigureHighlightParams* function.
- Fixed potential memory leak in the *OERenderMolecule* functions.

6.27.4 Documentation changes

- All images in this documentation were automatically regenerated to reflect the changes made since the previous release.

6.28 OEDepict TK 2.2.3

6.28.1 New features

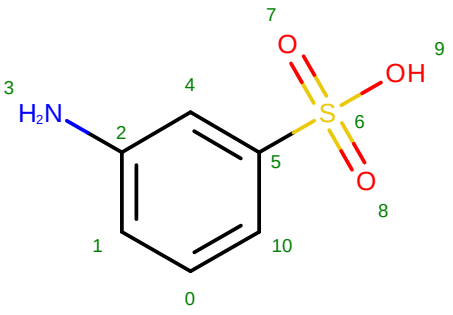
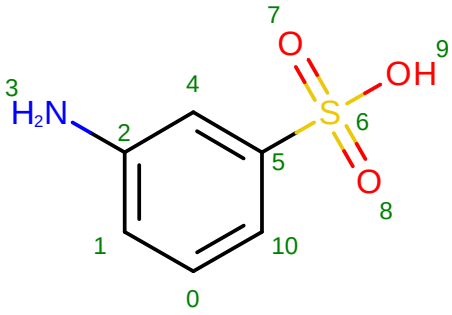
- *OEPrepareDepictionOptions* added to allow finer grained control of the *OEPrepareDepiction* function.
- *OEGetMoleculeCoords* and *OEGetDisplayCoords* functions added that convert 2D points between the molecule and the display coordinate systems.
- *OEAddHighlightOverlay* added to elegantly highlight overlapping sections of a molecule.
- Added the following functions to *OESystem* to provide predefined sets of colors:
  - `OEGetColors`
  - `OEGetContrastColors`
  - `OEGetDeepColors`
  - `OEGetLightColors`
  - `OEGetVividColors`
- Added the following colors to *OESystem* :
  - `OESystem.OEBrass`
  - `OESystem.OECopper`

- `OESystem.OEGold` (the original color was renamed `OESystem.OEMandarin`)
- `OESystem.OESilver`
- `OESystem.OEPewter`

6.28.2 Major bug fixes

- Fixed Linux thread safety when writing `.png` files. The Cairo graphics library (http://cairographics.org) used to draw `.png` images on Linux is not thread safe. **OEDepict TK** now performs its own mutual exclusion when calling into the Cairo graphics library to ensure thread safety. It is strongly recommended to use the `.svg` file format when speed and server throughput is a concern. Web browser support for `.svg` has improved significantly in recent years and it is more effective at representing molecule graphs since they are mostly vector graphics. The only time `.png` is preferable over `.svg` is when rendering heavy raster graphics like property maps from **Grapheme TK**.
- Fixed thread safety of creating `.svg` images.
- Increased the default font size of the atom label and atom property label. See the effect of this change in *Table: Example of increasing the default font size*.

Table 10: **Example of increasing the default font size**

OEDepict TK 2.2.2	OEDepict TK 2.2.3
	

- The reaction role of hydrogens will no longer be thrown away during the depiction process. Specifically, all information present on explicit hydrogens will be preserved during the depiction process.
- **OEDepict TK** generated `.pdf` files opened by several PDF viewers would show the file as edited due to an internal representation problem. PDF viewers should no longer modify **OEDepict TK** generated `.pdf` files.
- **OEDepict TK** generated `.pdf` files should no longer cause the following warning messages when read into Adobe Acrobat:

```
The font 'Arial-Bold' contains a bad /BBox.
```

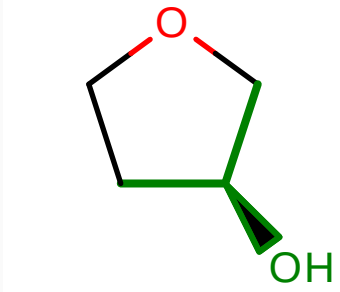
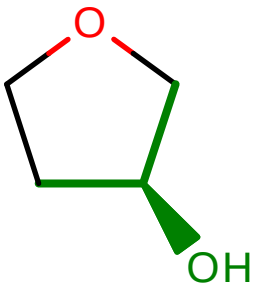
The Helvetica font will now be used as the default font in `.pdf` files. The *`OEFontFamily.Arial`* constant will now map to Helvetica as well in `.pdf` files.

6.28.3 Minor bug fixes

- Isotope hydrogens are never suppressed by the *`OEPrepareDepiction`* function.
- `.svg` and `.bsvg` image dimensions are now specified by the "viewBox" parameter rather than the "width" and "height" parameters. This allows scaling the images in web applications.

- Functions and constructors of classes of Ogham API will now throw warning messages about being deprecated.
- Removed unbounded stack allocations.
- Fixed the problem when wedge bonds were not highlighted correctly when using the *OEHighlightStyle.Color* highlighting style. See the effect of this change in *Table: Example of highlighting wedge bonds with "Color" style*.

Table 11: **Example of highlighting wedge bonds with "Color" style**

OEDepict TK 2.2.2	OEDepict TK 2.2.3
	

- Functions and methods that returned a *OE2DPoint* to Java and C# no longer return a dangling reference that was easy to cause crashes with.

6.28.4 Documentation changes

- Updated the *Highlighting Overlapped Patterns* section and added a new code example that shows how to highlight overlapping matches using the new *OEAddHighlightOverlay* function.
- Documentation updated to reflect that *OE2DMolDisplayOptions.SetAtomPropertyFunctor* and *OE2DMolDisplayOptions.SetBondPropertyFunctor* are preferred over using *OE2DAtomDisplay.SetProperty* and *OE2DBondDisplay.SetProperty* directly to ensure molecule scaling fully accounts for the size of the labels.
- All images in this documentation were automatically regenerated to reflect the changes made since the previous release.
- Added the missing documentation for the following *OESystem* colors:
  - `OESystem.OEGreenBlue`
  - `OESystem.OEGreenTint`
- The example for a python-based web interface and web service have been moved to our Python Cookbook.

6.29 OEDepict TK 2.2.2

6.29.1 New features

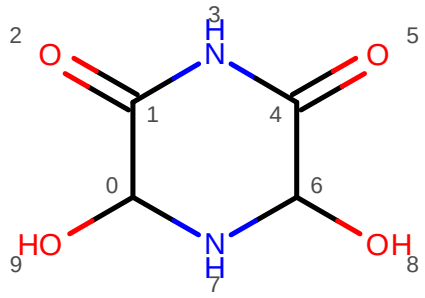
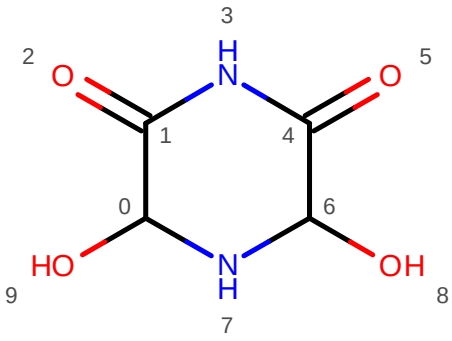
- *OEPrepareAlignedDepiction* can now take a OESubSearch or OEMCSSearch object to allow the layout to be driven by multiple atom mappings at the same time. The mapping that generates the best alignment will be returned in an *OEAlignmentResult* object.
- *OEGetDefaultAtomColor* added to return the default color associated with an atomic number for the given atom color style.

- *OEGreenBoxPen* and *OEGreenPen* pens added.
- *OEDepictCoordinates* is now deprecated. 2D coordinate generation is now included in `OEChem` as the `OEGenerate2DCoordinates` function. Please switch to using that function instead.

6.29.2 Major bug fixes

- Improved the positioning of atom properties by considering the position of implicit hydrogen labels. See the effect of this change in *Table: Example of improving the positioning of atom properties*.

Table 12: **Example of improving the positioning of atom properties**

OEDepict TK 2.2.1	OEDepict TK 2.2.2
	

6.29.3 Minor bug fixes

- **OEDepict TK** will automatically render enhanced stereo by default now. Previously, the *OEConfigure2DMolDisplayOptions* function would not automatically enable rendering enhanced stereo information. The behavior can be changed by the *OEAtomStereoStyle.Display.MDLStereoCenter* and *OE2DMolDisplaySetup.AtomStereoStyle* constants.
- Increased the gap between the bond end and the atomic label for iodine (atomic label "I").
- Added "BRIEF" line to interfaces of **OEDepict TK** examples.
- *OEWriteImage* and *OERenderMolecule* that take an oeostream no longer call the stream's `close` method. It is the responsibility of the caller to call `close` on the stream if it is appropriate.

6.29.4 Documentation changes

- All images in this documentation were automatically regenerated to reflect the changes made since the previous release.

6.30 OEDepict TK 2.2.1

6.30.1 New features

- Added cubic Bézier drawing method to the following classes:
  - *OEImageBase*
  - *OEImage*
  - *OEImageFrame*

- Added quadratic Bézier drawing method to the following classes:
  - *OEImageBase*
  - *OEImage*
  - *OEImageFrame*
- Added rotation angle property to the *OEFont* class that allows to render text in various angles when calling the *OEImageBase.DrawText* method.
  - *OEFont.GetRotationAngle*
  - *OEFont.SetRotationAngle*

6.30.2 Major bug fixes

- When aligning a molecule to a template, atom clashes between the template atoms are ignored.
- *OEPrepareDepiction* will now suppress hydrogens even when 2D coordinates are not generated, i.e., `clearcoords = false`.

6.30.3 Minor bug fixes

- Highlighting a single atom with the *OEHighlightStyle.Stick* will now work if that atom is bonded to a highlighted atom.
- Fixed problem when the scaling of small molecules resulted in depicting atom labels outside the canvas.
- Fixed problem that incorrectly positioned the "+" sign between reaction components introduced by the last **OEDepict TK** release, 2.2.0.

6.30.4 Documentation changes

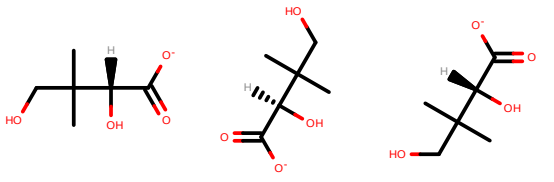
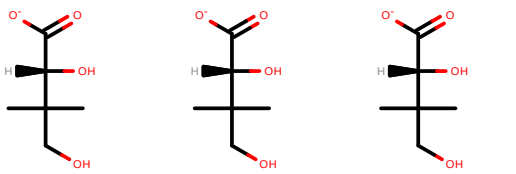
- All images in this documentation were automatically regenerated to reflect the changes made since the previous release.

6.31 OEDepict TK 2.2.0

6.31.1 New features

- *OEPrepareDepiction* now generates canonical depiction coordinates, i.e., the same 2D coordinates are generated regardless of the atom and bond ordering in the molecule. See the effect of this change in *Table: Example of default element color changes*.

Table 13: **Example of generating depiction coordinates by the OEPrepareDepiction function**

OEDepict TK 2.1.0	OEDepict TK 2.2.0
	

- Added the following non-linear color gradient to OESystem:

  - OEExponentColorGradient
  - OEExponentialColorGradient
  - OELogarithmicColorGradient

  These classes along with the OELinearColorGradient class now derive from the OEColorGradientBase base class.

- *OEAddHighlighting* function overloads added that take both atom and bond predicates.
- *OEAlignmentOptions.SetFixedCoords* method added that forces the *OEPrepareAlignedDepiction* function to use the coordinates in the reference molecule instead of depicting them from scratch.
- *OE2DMolDisplayOptions.SetBondLineGapScale* method added for changing the gap between the lines of double and triple bonds.
- *OE2DMolDisplayOptions.SetBondLineAtomLabelGapScale* method added for changing the gap between the atom labels and the end of the bond lines.

6.31.2 Minor bug fixes

- *OEStipple.NoLine* and *OEFill.On* pen fixed on Windows when drawing a pie.
- Refined the method that is used to calculate the gap between the atom labels and the lines of the bonds when there is a charge or there are more than 3 explicit hydrogen neighbors.
- Additional warning messages are thrown when either the `OE3DToBondStereo` or the `OE3DToAtomStereo` functions fail during *OEPrepareDepiction*.

6.31.3 Documentation changes

- All images in the documentation were automatically regenerated to reflect the changes made since the previous release.
- Added several code snippets to the *OERenderMolecule* function.

6.32 OEDepict TK 2.1.0

6.32.1 New features

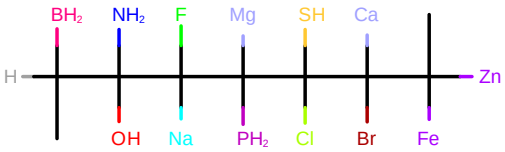
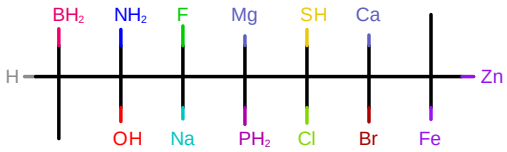
- Added a new highlighting style, *OEHighlightStyle.Cogwheel*, and the corresponding highlighting class *OEHighlightByCogwheel*.
- Added accessor methods for many parameters to the following highlighting classes:
  - *OEHighlightByBallAndStick*
  - *OEHighlightByColor*
  - *OEHighlightByStick*
- Added *OE2DMolDisplayOptions.SetBondWidthScaling* method that allows line width of the bonds to be increased or decreased based upon the molecule scaling factor.
- Added the following classes and functions to provide a convenient way to generate multi-page reports.
  - *OEReportOptions* and *OEReport* classes
  - *OEConfigureReportOptions* and *OESetupReportOptions* functions
  - *OEWriteReport* function
- Implemented the following Java **OEDepict TK** examples:

  - *Mols2PDF*
  - *ViewMDLSearch*
  - *SMARTSAlign2D*
- Added the `OEMultiPageImageFile.GetPage` method that provides access to the individual pages of a multi-page image file.
- Added the `OEConfigureImageGridNumColumns` and `OEConfigureImageGridNumRows` functions to take the default number of columns and rows.
- `OEPrepareAlignedDepiction` has been improved the generate better 2D molecule alignments by:
  - Allowing rotations around unspecified cis/trans double bonds
  - Refining the method that scores alignments based on internal atom clashes

6.32.2 Major bug fixes

- Added darker default colors for some of the elements for the `OEAtomColorStyle.WhiteCPK` color scheme. See the effect of these changes in *Table: Example of default element color changes*.

Table 14: **Example of default element color changes**

OEDepict TK 2.0.4	OEDepict TK 2.1.0
	

6.32.3 Minor bug fixes

- Fixed the coordinate system to start from the top-left corner for raster images on OS X when using the `OEImageBase.DrawPoint` method. This was causing images to appear flipped around the y-axis.
- The colors for `OEGetHighlightColor` were sometimes not initialized properly due to undefined library initialization order. This was causing the function to always return white.
- Atoms that have both the atomic number and the atom map index of zero will now be rendered as *.
- Synchronized code examples in all four supported languages (C++, Python, Java and C#) to ensure that they have the same behavior and generate identical images.
- `OEPrepareDepiction` and `OEDepictCoordinates` could crash or generate random coordinates for molecules with adjacent spiro centers.

6.32.4 Documentation changes

- Added the *Multi Page Reports* section to document the usage of the *OEReport* object.
- All images in the documentation were automatically regenerated to reflect the changes made since the previous release.

6.33 OEDepict TK 2.0.4

6.33.1 New features

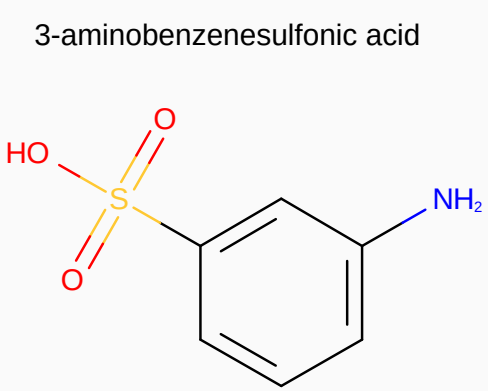
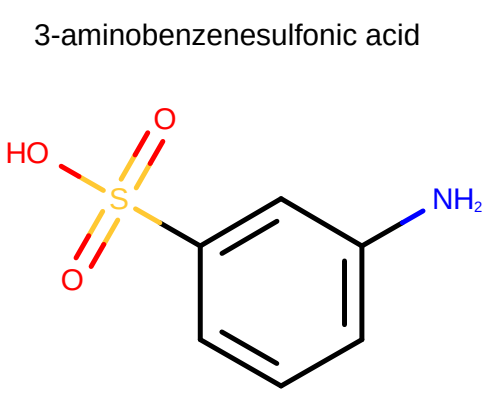
- Pie drawing, semi-circle, methods added to the following classes:
  - *OEImageBase*
  - *OEImage*
  - *OEImageFrame*
- An example of a python-based web interface and web service have been added. The code for this example was on GitHub, as it required multiple files.
- Enabling the depiction of MDL's enhanced stereochemical representation by default. See the *OEAtomStereoStyle.Display.MDLStereoCenter* constant.

6.33.2 Major bug fixes

- The following molecule depiction defaults have been changed:
  - The background color is now pure white instead of a very light grey.
  - The bond width is now 2.0 instead of 1.0.

  See the effect of these changes in *Table: Example of default molecule depictions*.

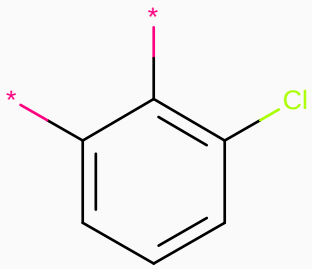
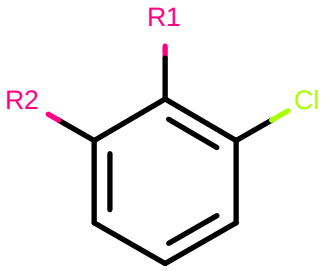
Table 15: **Example of default molecule depictions**

OEDepict TK 2.0.3	OEDepict TK 2.0.4
	

6.33.3 Minor bug fixes

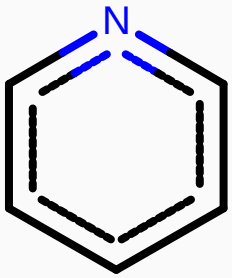
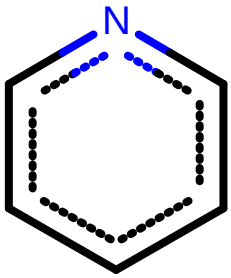
- Adding the depiction of R-groups for molecules that have not been initialized from a MDL query file. See the example the in the *Table: Example of depicting R-Groups*.

Table 16: **Example of depicting R-Groups**

OEDepict TK 2.0.3	OEDepict TK 2.0.4
[R1]c1c(cccc1Cl)[R2] 	[R1]c1c(cccc1Cl)[R2] 

- Fixed a bug that would erroneously hide the following types of hydrogens in a depiction:
  - isolated hydrogens
  - charged hydrogens
  - hydrogen isotopes
- Fused ring system templates added for 29 different types of complex depiction cases.
- *OE2DAtomDisplay.SetHCount* method will now place the hydrogen position correctly as returned by *OE2DAtomDisplay.GetHPosition*, i.e., the position relative to the heavy atom. An incorrect position was possible when generating 2D coordinates with *OEPrepareDepiction* with the parameter `suppressH` set to `false`.
- The stipple factor of a line in now based on the line width instead of the scaling factor of the molecule display.

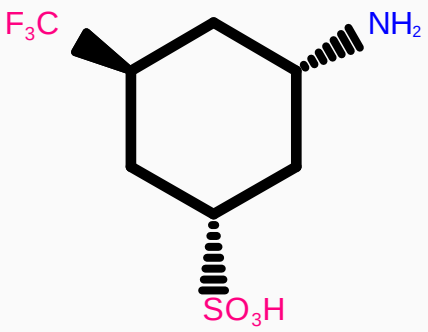
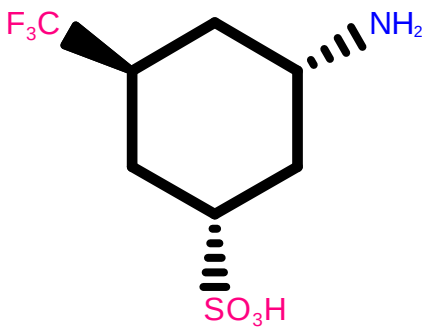
Table 17: **Example of line width affecting the stipple factor**

OEDepict TK 2.0.3	OEDepict TK 2.0.4
	

- Atom stereo depiction improved:
  - The number of individual lines used in the *OEBondDisplayType.Hash* and the *OEBondDisplayType.Wavy* stereo bonds now depends on the bond width.
  - Heuristic used to shorten hash and wedge stereo bonds improved.

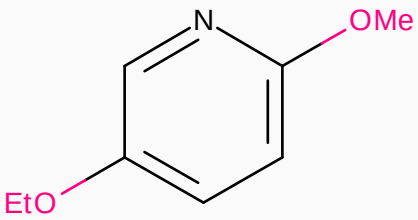
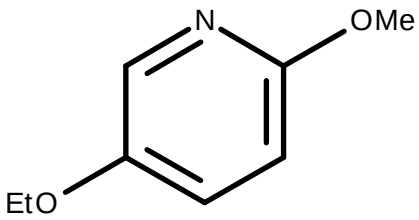
See the effect of these improvements in *Table: Example of depicting atom stereo*.

Table 18: **Example of depicting atom stereo**

OEDepict TK 2.0.3	OEDepict TK 2.0.4
	

- `OEBondDisplayType.Any` bond type will now properly set the stipple factor for the end pen of that bond.
- Fixing the problem when the default atom label colors were not updated after atom color style change. For example when setting the atom color style to `OEAtomColorStyle.WhiteMonochrome` the super atoms remained pink rather than changed to black. See the effect of fix in *Table: Example of using the 'WhiteMonochrome' atom color style*.

Table 19: **Example of using the 'WhiteMonochrome' atom color style**

OEDepict TK 2.0.3	OEDepict TK 2.0.4
	

6.33.4 Documentation changes

- All images are automatically regenerated to reflect the changes made since the previous release.

6.34 OEDepict TK 2.0.3

6.34.1 New features

- `OEDrawCurvedArrow` function added.

- *OEPrepareAlignedDepiction* now allows more advanced customization of the alignment process through the *OEAlignmentOptions* class.
- **OEChem TK's** new RMSD method is used in the molecule alignment process to improvement alignment performance.

6.34.2 Major bug fixes

- *OEPrepareAlignedDepiction* performance improvement of 10% on average.
- Windows png images have been improved in the following ways:
  - Dash patterns for line drawing corrected.
  - Gamma correction turned off when colors are blended with background colors. As a result the generated png images look sharper.
  - Text anti-aliasing methodology is now dependent on the font size to ensure fonts look good at all sizes. The method of anti-aliasing changes when increasing the font size from 8 to 9.
- The following atom label layout changes:
  - If the atom has a charge, then it is rendered as a super-script on the right side of the atomic label. In the case when the implicit hydrogen symbol is placed on the right side of the atom, then the charge also follows the hydrogen symbol. (See section GR 5.1 in [Brecher-2008])
  - The dot(s) representing the radical atom follows the atoms' element symbol on the left side. In the case when the implicit hydrogen symbol is placed on the right side of the atom, then the radical dot(s) also follows the hydrogen symbol. (See section GR 5.3 in [Brecher-2008])
- Atom labels clashing with bonds have been improved by shortening the length of the depicted bond in certain cases.
- MDL query molecule atom label rendering has been improved.
- *OEDepictCoordinates* clash detection has been improved to avoid unnecessary stretching of the bonds of explicit hydrogens. A tolerance has been introduced that allows smaller distances between hydrogen atoms.

6.34.3 Minor bug fixes

- *OE2DMolDisplayOptions.SetAtomLabelFontScale* upper limit has been increased from *2.0* to *3.0*.
- New fused ring system templates added.
- Atom and bond highlighting now scales with the molecule instead of the dimension of the image.
- Aromatic circles now ignore the fill style of the circle.
- The following methods will now throw warnings if given nonsensical arguments:
  - *OEImage.DrawCircle*
  - *OEImage.DrawArc*
  - *OEImage.DrawLine*
  - *OEImage.DrawPolygon*
  - *OEImage.DrawRectangle*
  - *OEImage.DrawText*
  - *OEImage.DrawTriangle*

  This should make drawing mistakes appear earlier in development, instead of only when they are rendered to a specific file format.

- *OE2DMolDisplay.GetAtomDisplay* and *OE2DMolDisplay.GetBondDisplay* no longer incorrectly check the atom and bond indices.
- *OE2DMolDisplay* constructors now throw a warning when a molecule being depicted has 3D coordinates and throw an error if the molecule has no coordinates.
- *OEImage.Clear* is now called when constructing a new *OEImage* object.

6.34.4 Documentation changes

- All images are automatically regenerated to reflect the changes made since the previous release.

6.35 OEDepict TK 2.0.2

6.35.1 New features

- Adding a scaling factor that can be used to increase or decrease the size of the fonts used to depict atom properties. (*OE2DMolDisplayOptions.SetAtomPropLabelFontScale*)
- Adding a scaling factor that can be used to increase or decrease the size of the fonts used to depict bond properties. (*OE2DMolDisplayOptions.SetBondPropLabelFontScale*)
- Adding the following examples:
  - *Depict.java*
  - *Match2Img.java*
  - *MCSAlign2D.java*
  - *MDLQuery2Img.java*
  - *MDLReaction2Img.java*
  - *Mol2Img.java*
  - *Mols2Img.java*
- The constructor of the *OEImage* class now takes a background color that is used to clear the image upon construction. A `.png` image with transparent background can be generated by using the `OETransparentColor` color as a background color when constructing an *OEImage* object.
- Fixing the run-time initialization of the color that is used to depict super atoms.

6.35.2 Major bug fixes

- The cairo graphics library (http://cairographics.org) used to draw `.png` images on Linux has a bug where it is not thread safe when rendering text. **OEDepict TK** now performs its own mutual exclusion when calling into the cairo graphics library for text rendering to ensure thread safety.
- Improving the depiction of super atoms (See examples in the *OESuperAtomStyle* namespace)

6.35.3 Minor bug fixes

- *OERenderMolecule* called with an image and a molecule now uses the width and height of the image being passed.
- Fixing a problem where isolated atoms were not highlighted when using the *OEHighlightStyle.Color* and *OEHighlightStyle.Stick* highlighting styles.
- *OEDepictCoordinates* throws a warning when trying to generate 2D coordinates for a two member ring, for example, `C1N1`. The function will then fall back to trying to generate coordinates for those atoms as a chain.

- The `OEDisassembleExpressions` function tags atoms and bonds of the query molecule that can later be used by the *OE2DMolDisplay* constructor when creating atom and bond displays.
- Throwing a warning when attempting to draw a line or an arc with the *OEStipple.NoLine* pen property. Throwing a warning when attempting to draw a rectangle, circle, triangle or a polygon with *OEStipple.NoLine* and *OEFill.Off* pen properties.
- Making sure that no radical dots are displayed for a query atom that has implicit hydrogen count information.
- Before registering an aromatic ring for the *OEAromaticStyle.Circle* style, a check is performed to ensure that the detected ring is not a fused ring of two smaller rings.
- Making the background color consistent for the monochrome atom color styles. (See: *OEAtomColorStyle.WhiteMonochrome* and *OEAtomColorStyle.BlackMonochrome*)
- When two molecules are aligned by calling the *OEPrepareAlignedDepiction* function, the fit molecule is scaled by the average bond length of the reference molecule.

6.35.4 Deprecated Ogham

- Deprecating the `OEDepictFixedCoordinates` function. For molecule alignment, please use the *OEPrepareAlignedDepiction* function.

6.35.5 Documentation changes

- Renaming the Application chapter to *OEDepict Examples Summary* and inserting the source code of the examples into the *Examples* chapter.
- Adding code snippets for the *OEAddHighlighting* function.
- Describing the layout of a depicted atom in the *OE2DAtomDisplay* class.
- All images in this manual are now automatically generated as part of the release process and tested against benchmark images on all supported platforms.

6.36 OEDepict TK 2.0.1

6.36.1 Minor bug fixes

- On creation of an *OE2DMolDisplay* object its scale is adjusted by the average bond length of the original 2D coordinates of the molecule. This affected any rendering where the 2D coordinates are not generated by OEDepict.
- Speed optimization of the *OEPrepareAlignedDepiction* function were added.

6.37 OEDepict TK 2.0.0

6.37.1 New features

- Using *OE2DPoint* that represents coordinates
- Using OEColor that handles transparency
- Adding OELinearColorGradient class that allows color ramping
- **Supporting various font properties (*OEFont*)**
  - size
  - color
  - style (italic, bold) (*OEFontStyle*)

  - family (Arial, Helvetica, etc.) (*OEFontFamily*)

- **Supporting various line properties (*OEPen*)**
  - line width
  - background, foreground color
  - various stipple styles (*OEStipple*)
  - standard line join, line cup styles (*OELineJoin*, *OELineCap*)
- **Providing convenient molecule depiction that is highly customizable (*OE2DMolDisplayOptions*)**
  - different aromatic ring representations (*OEAromaticStyle*)
  - atom coloring (monochrome, CPK) (*OEAtomColorStyle*)
  - bond coloring (atom-based, monochrome) (*OEBondColorStyle*)
  - various atom stereo representations (*OEAtomStereoStyle*)
  - various bond stereo representations (*OEBondStereoStyle*)
  - various explicit/implicit hydrogen representations (*OEHydrogenStyle*)
  - depicting atom properties (*OEDisplayAtomPropBase*)
  - depicting bond properties (*OEDisplayBondPropBase*)
  - setting atom label, atom and bond property label font styles
  - various super atom styles (*OESuperAtomStyle*)
- Depicting atom radicals (*OERadicalDisplayType*)
- Adding the *OEPrepareDepiction* function that prepares a molecule before depiction.
- Adding the *OEPrepareAlignedDepiction* function that generates 2D-coordinates of a molecule to maximize its alignment to a reference molecule.
- Introducing the *OEImageFrame* and the *OEImageGrid* classes that allows various molecule layouts (see the *Molecule Layouts* chapter)
- Implementing three customizable highlighting styles and allowing to utilize user-defined ones (see also the *Highlighting* chapter) - ball and stick (*OEHighlightByBallAndStick*) - color (*OEHighlightByColor*) - stick (*OEHighlightByStick*)
- Adding the *OEAddHighlighting* function that helps to highlight atoms and/or bonds
- Implementing three arrow drawing functions: * *OEDrawEquilibriumArrow* * *OEDrawReactionArrow* * *OEDrawResonanceArrow*
- Adding functions that automatically generate standard interfaces for OEDepict applications: - *OEConfigure2DMolDisplayOptions* - *OEConfigureHighlightParams* - *OEConfigureImageGridParams* - *OEConfigureMultiPageParams*
- Adding various rendering functions (*OERenderMolecule*)
- Adding functions to write images (*OEWriteImage*)
- Supporting the following image file formats:

Graphics File Format	File Extension
PNG (Portable Network Graphics)	`.png`
SVG (Scalable Vector Graphics)	`.svg`
bare SVG (with no header)	`.bsvg`
Postscript	`.ps`
Encapsulated PostScript	`.eps`
PDF (Portable Document Format)	`.pdf`

Note

PNG support is based upon the cairo graphics library on Linux. Most linux distributions install this shared library by default, but it may be possible for a super stripped down installation to not include it. **OEDepict TK** has no specific version requirement for this library, so the *cairo* package can be installed through traditional distribution specific means: rpm, yum, yast, zypper, etc.

- Supporting multi-page images (see the *Multi Page Image* chapter)
- Supporting to depict MDL queries (see the *MDL Query Depiction* chapter)
- Supporting MDL reaction depiction (see the *MDL Reaction Depiction* chapter)
- Providing more than 40 example codes in four different languages (C++, Python, Java, C#) that demonstrate the functionalities of the the new toolkit.
- Providing 10 C++ applications (see the *OEDepict Examples Summary* chapter)

6.37.2 Deprecated Ogham

The following classes, functions and namespaces are **not** going to be supported any longer and will be deprecated in the future:

Deprecated Ogham classes:

- `OE8BitImage`
- `OEAtomStyle`
- `OEBondStyle`
- `OEDepictBase`
- `OEDepictColor`
- `OEDepictView`
- `OEPSImage`
- `OEPoly`
- `OEPolyF`
- `OESVGImage`
- `OEVert`
- `OEVertF`

Deprecated Ogham functions:

- `OEWriteBMP`

- OEWriteEPS
- OEWriteGIF
- OEWritePPM
- OEWriteRGB
- OEWriteXPM

Deprecated Ogham namespace:

- OENamedColor

CHAPTER

SEVEN

COPYRIGHT AND TRADEMARKS

© 1997-2024 Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA. All rights reserved.

OpenEye, Lexichem, ROCS, Grapheme, and Orion are registered trademarks of Cadence Design Systems, Inc.

This material contains proprietary information of Cadence Design Systems, Inc. Use of copyright notice is precautionary only and does not imply publication or disclosure.

The information supplied in this document is believed to be true, but no liability is assumed for its use or the infringement of the rights of others resulting from its use. Information in this document is subject to change without notice and does not represent a commitment on the part of Cadence Design Systems, Inc.

This package is sold/licensed/distributed subject to the condition that it shall not, by way of trade or otherwise, be lent, re-sold, hired out, or otherwise circulated without Cadence Design Systems' prior consent, in any form of packaging or cover other than that in which it was produced. No part of this manual or accompanying documentation, may be reproduced, stored in a retrieval system on optical or magnetic disk, tape, CD, DVD or other medium, or transmitted in any form or by any means, electronic, mechanical, photocopying recording, or otherwise for any purpose other than for the purchaser's personal use without a legal agreement or other written permission granted by Cadence.

This product should not be used in the planning, construction, maintenance, operation or use of any nuclear facility nor the flight, navigation, or communication of aircraft or ground support equipment. Cadence Design Systems, Inc., shall not be liable, in whole or in part, for any claims arising from such use, including death, bankruptcy or outbreak of war.

Windows is a registered trademark of Microsoft Corporation. Apple, OS X, macOS, and Macintosh are registered trademarks of Apple Inc. UNIX is a registered trademark of the Open Group. Linux is a registered trademark of Linus Torvalds. Red Hat is a registered trademark of Red Hat, Inc. SUSE, SLED, and SLES are registered trademarks of Novell, Inc. Ubuntu is a registered trademark of Canonical Ltd.

SYBYL is a registered trademark of Certara, L.P. MDL, BIOVIA and ISIS are registered trademarks of Dassault Systemes and/or its affiliates. SMARTS and SMIRKS are registered trademarks of Daylight Chemical Information Systems, Inc. MacroModel is a registered trademark of Schrodinger, LLC.

Python is a registered trademark of the Python Software Foundation. Django is a registered trademark of the Django Software Foundation. Java is a registered trademark of Oracle and/or its affiliates.

Other products and software packages referenced in this document are trademarks and registered trademarks of their respective vendors or manufacturers.

CHAPTER

EIGHT

SAMPLE CODE

For all of the code examples and samples within documents published by OpenEye, Cadence Molecular Sciences, the following terms of use apply.

TERMS FOR USE OF SAMPLE CODE

The software below ("Sample Code") is provided to current licensees or subscribers of Cadence products or SaaS offerings (each a "Customer"). Customer is hereby permitted to use, copy, and modify the Sample Code, subject to these terms. Cadence claims no rights to Customer's modifications. Modification of Sample Code is at Customer's sole and exclusive risk. Sample Code may require Customer to have a then current license or subscription to the applicable Cadence offering.

THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. CADENCE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be liable for any damages or liability in connection with the Sample Code or its use.

CHAPTER

NINE

CITATION

Please follow the guidelines in this document to cite use of our products in research publications and other documents.

For more information, please contact OpenEye Support.

9.1 Orion®

To cite use of Orion, please use the following:

```
Orion <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

For example:

```
Orion 2025.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.eyesopen.com.
```

The version number for Orion appears in the product's release notes, for example, https://docs.eyesopen.com/orion-user-interface/releasenotes/index.html.

9.2 Orion Floes

To cite use of an Orion-based floe package, please use the following:

```
OpenEye <package-name> <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe,␣
↪NM. http://www.eyesopen.com.
```

For example:

```
OpenEye Large-Scale Floes 4.2.3. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

The version number for a floe package is displayed on the first page of the package's release notes. For example: https://docs.eyesopen.com/floe/modules/large_scale_floes/docs/source/releasenotes_2025_1.html.

9.3 Toolkits and Applications

To cite OpenEye toolkits or applications used in your work, please use the following:

```
OpenEye Toolkits [or Applications] <version-number>. OpenEye, Cadence Molecular Sciences,
↪ Santa Fe, NM. http://www.eyesopen.com.
```

For example:

```
OpenEye Toolkits 2024.2.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

The Applications and Toolkits are released as a package, multiple times per year. The first part of the version number designates the calendar year. The version number appears on the summary document for the package.

If you want to cite an individual application or toolkit, you can use the syntax:

```
<product name> <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

where <*product name*> is:

- Bioisostere TK
- OEChem TK
- OEDepict TK
- OEDocking TK
- OEFF TK
- Eon TK
- FastROCS TK
- Grapheme TK
- GraphSim TK
- Lexichem TK
- OEMedChem TK
- MolProp TK
- Omega TK
- Quacpac TK
- Shape TK
- Spicoli TK
- Spruce TK
- Szmap TK
- Szybki TK
- Zap TK
- AFITT Application
- BROOD Application
- OEDocking Application
- EON Application
- OMEGA Application
- PICTO Application
- pKa-Prospector Application
- QUACPAC Application

- ROCS Application
- SiteHopper Application
- SPRUCE Application
- SZMAP Application
- SZYBKI Application
- VIDA Application

You can use the version number of the individual Application or Toolkit. For example:

```
SZYBKI Application 2.8.0.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://
↪www.eyesopen.com.

Szybki TK 2.8.0.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

The first documentation pages give the product names and version numbers, as in https://docs.eyesopen.com/applications/szybki/index.html and https://docs.eyesopen.com/toolkits/python/szybkitk/index.html.

9.4 Publications for Bibliographies

Please also cite these publications if the corresponding product was used in your research.

9.4.1 Orion

Sørensen, J., Bannan, C.C., Calabrò, G., Jain, V., Ovanesyan, G., Smith, A., Zhang, S., Bayly, C.I., Darden, T.A., Geballe, M.T., LeBard, D.N., McGann, M., Moon, J.B., Muddana, H.S., Shewmaker, A., LaFon, J., Tolbert, R.W., Skillman, A.G., Nicholls, A. Orion® A Cloud-Native Molecular Design Platform. In *Computational Drug Discovery*. V. Poongavanam and V. Ramaswamy, Eds. Wiley-VCH, 2024; pp. 579–615. DOI: 10.1002/9783527840748.ch24

9.4.2 AFITT and FLYNN Applications

Wlodek, S.; Skillman, A.G.; Nicholls, A. Automated Ligand Placement and Refinement with a Combined Force Field and Shape Potential. *Acta Crystallogr., Sect. D: Struct. Biol.* **2006**, *62* (7), 741–749. DOI: 10.1107/s0907444906016076

9.4.3 OEDocking Application and Toolkit

Kelley, B.P.; Brown, S.P.; Warren, G.L.; Muchmore, S.W. POSIT: Flexible Shape-Guided Docking For Pose Prediction. *J. Chem. Inf. Model.* **2015**, *55* (8), 1771-1780. DOI: 10.1021/acs.jcim.5b00142

McGann, M. FRED Pose Prediction and Virtual Screening Accuracy. *J. Chem. Inf. Model.* **2011**, *51* (3), 578-596. DOI: 10.1021/ci100436p

McGann, M. FRED and HYBRID Docking Performance on Standardized Datasets. *J. Comput.-Aided Mol. Des.* **2012**, *26* (8), 897-906. DOI: 10.1007/s10822-012-9584-8

9.4.4 OMEGA Application and Toolkit

If you are using OMEGA in torsion driving mode, please cite the following two publications.

Hawkins, P.C.D.; Skillman, A.G.; Warren, G.L.; Ellingson, B.A.; Stahl, M.T. Conformer Generation with OMEGA: Algorithm and Validation Using High Quality Structures from the Protein Databank and the Cambridge Structural Database. *J. Chem. Inf. Model.* **2010**, *50* (4), 572-584. DOI: 10.1021/ci100031x

Hawkins, P.C.D. and Nicholls, A. Conformer Generation with OMEGA: Learning from the Data Set and the Analysis of Failures. *J. Chem. Inf. Model.* **2012**, *52* (11), 2919–2936. DOI: /10.1021/ci300314k

If you are using OMEGA in macrocyle mode, please cite the following.

Hawkins, P.C.D. and Wlodek, S. Decisions with Confidence: Application to the Conformation Sampling of Molecules in the Solid State. *J. Chem. Inf. Model.* **2020**, *60* (7), 3518–3533. DOI: /10.1021/acs.jcim.0c00358

9.4.5 ROCS Application

Hawkins, P.C.D.; Skillman, A.G.; Nicholls, A. Comparison of Shape-Matching and Docking as Virtual Screening Tools. *J. Med. Chem.* **2007**, *50* (1), 74-82. DOI: 10.1021/jm0603365

9.5 OpenEye MMDS Web Service

To cite use of the Macromolecular Data Service (MMDS) web service, please use the syntax:

```
Macromolecular Data Service <version-number>. OpenEye, Cadence Molecular Sciences, Santa␣
↪Fe, NM. http://www.eyesopen.com.
```

For example:

```
Macromolecular Data Service 1.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

The MMDS version number appears on the web service's release notes.

CHAPTER

TEN

TECHNOLOGY LICENSING

OpenEye products use the following technology under license.

Some of the open source technologies we use are licensed under the GNU Public License (GPL) version 2 or 3. In all instances, these technologies are communicated with using the "at arms length" principle, using pipes and command line arguments.

While the software is in some instances assembled into a container, that assembly does not invoke the GPL copyleft scope. For each Floe package where these are used, the environment log clearly details the installed programs and their version numbers. With the links below for each technology, the source for each of those programs is available.

Table 1: Licensed Technology

Name of Project	Website	License URL
_libgcc_mutex	https://github.com/conda-forge/_libgcc_mutex-feedstock	https://github.com/conda-forge/_libgcc_mutex-feedstock/blob/master/LICENSE.txt
_openmp_mutex	https://github.com/conda-forge/_openmp_mutex-feedstock	https://github.com/conda-forge/_openmp_mutex-feedstock/blob/main/LICENSE.txt
ablang2	https://github.com/TobiasHeOl/AbLang2	https://github.com/TobiasHeOl/AbLang2/blob/main/LICENSE
abseil-cpp	https://github.com/abseil/abseil-cpp	https://github.com/abseil/abseil-cpp/blob/master/LICENSE
absl-py	https://github.com/abseil/abseil-py	https://github.com/abseil/abseil-py/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
aiohappyeyeballs	https://github.com/aio-libs/aiohappyeyeballs	https://github.com/aio-libs/aiohappyeyeballs/blob/main/LICENSE
aiohttp	https://docs.aiohttp.org/en/stable/	https://github.com/aio-libs/aiohttp/blob/master/LICENSE.txt
aiosignal	https://github.com/aio-libs/aiosignal	https://github.com/aio-libs/aiosignal/blob/master/LICENSE
Amazon Linux 2	https://aws.amazon.com/amazon-linux-2	N/A
AmberUtils	http://ambermd.org	N/A
ambit	https://github.com/khimaros/ambit	https://github.com/khimaros/ambit/blob/master/LICENSE
amqp	https://github.com/celery/py-amqp	https://github.com/celery/py-amqp/blob/master/LICENSE
anaconda-anon-usage	https://github.com/anaconda/anaconda-anon-usage	https://github.com/anaconda/anaconda-anon-usage/blob/main/LICENSE
angular	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md
angular-animate	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md
angular-cache	https://github.com/jmdobry/angular-cache	https://github.com/jmdobry/angular-cache/raw/master/LICENSE
angular-cookies	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
angular-loggly-logger	https://github.com/ajbrown/angular-loggly-logger	https://github.com/ajbrown/angular-loggly-logger/raw/master/LICENSE.md
angular-mocks	https://github.com/angular/angular.js	https://raw.githubusercontent.com/angular/angular.js/master/LICENSE
angular-resource	https://github.com/angular/angular.js	https://raw.githubusercontent.com/angular/angular.js/master/LICENSE
angular-toggle-switch	https://github.com/cgarvis/angular-toggle-switch	https://github.com/cgarvis/angular-toggle-switch/blob/master/LICENSE
angular-ui-grid	https://github.com/angular-ui/ui-grid	https://github.com/angular-ui/ui-grid
angular-ui-router	https://github.com/angular-ui/ui-router	https://github.com/angular-ui/ui-router/raw/master/LICENSE
angular-ui-tree	https://github.com/angular-ui-tree/angular-ui-tree	https://github.com/angular-ui-tree/angular-ui-tree/raw/master/LICENSE
angular-vs-repeat	https://github.com/kamilkp/angular-vs-repeat	https://raw.githubusercontent.com/kamilkp/angular-vs-repeat/master/LICENSE
aniso8601	https://bitbucket.org/nielsenb/aniso8601	https://bitbucket.org/nielsenb/aniso8601/src/master/LICENSE
annotated-types	https://github.com/annotated-types/annotated-types	https://github.com/annotated-types/annotated-types/blob/main/LICENSE
antlr4-python3-runtime	http://www.antlr.org OR https://pypi.org/project/antlr4-python3-runtin	https://github.com/conda-forge/antlr4-python3-runtin/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
anyio	https://github.com/agronholm/anyio	https://github.com/agronholm/anyio/blob/master/LICENSE
appdirs	http://github.com/ActiveState/appdirs	http://github.com/ActiveState/appdirs/blob/master/LICENSE.txt
appengine	https://google.golang.org/appengine	https://github.com/golang/appengine/blob/master/LICENSE
arabic-reshaper	https://github.com/mpcabd/python-arabic-reshape	https://github.com/mpcabd/python-arabic-reshape/blob/master/LICENSE
archspec	https://github.com/archspec/archspec/blob/master/README.md	https://github.com/archspec/archspec/blob/master/COPYRIGHT
argon2-cffi	https://github.com/hynek/argon2-cffi	https://github.com/hynek/argon2-cffi/blob/master/LICENSE
argon2-cffi-bindings	https://github.com/hynek/argon2-cffi-bindings	https://github.com/hynek/argon2-cffi-bindings/blob/master/LICENSE
arpack	https://www.caam.rice.edu/software/ARPACK/	https://www.caam.rice.edu/software/ARPACK/
arrow	https://github.com/arrow-py/arrow	https://github.com/arrow-py/arrow?tab=Apache-2.0-1-ov-file#readme
ascii85	https://github.com/huandu/node-ascii85	https://github.com/huandu/node-ascii85/raw/master/LICENSE
ase	https://wiki.fysik.dtu.dk/ase/	https://wiki.fysik.dtu.dk/ase/development/licenseinfo.html#license-info
asgiref	https://github.com/django/asgiref/	https://github.com/django/asgiref/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
asn1crypto	https://github.com/wbond/asn1crypto	https://github.com/wbond/asn1crypto/blob/master/LICENSE
assertions go-render	https://github.com/smartystreets/assertions/internal/go-render/render	https://github.com/smartystreets/assertions/blob/master/internal/go-render/LICENSE
assertions oglmatchers	https://github.com/smartystreets/assertions/internal/oglematchers	https://github.com/smartystreets/assertions/blob/master/internal/oglematchers/LICENSE
assertions	https://github.com/smartystreets/assertions	https://github.com/smartystreets/assertions/blob/master/LICENSE.md
asttokens	https://github.com/gristlabs/asttokens	https://github.com/gristlabs/asttokens/blob/master/LICENSE
astunparse	https://github.com/simonpercivall/astunparse	https://github.com/simonpercivall/astunparse/blob/master/LICENSE
async-lru	https://github.com/aio-libs/async-lru	https://github.com/aio-libs/async-lru/blob/master/LICENSE
async-timeout	https://github.com/aio-libs/async-timeout	https://github.com/aio-libs/async-timeout/blob/master/LICENSE
atk-1.0	https://docs.gtk.org/atk/	https://gitlab.gnome.org/GNOME/atk/-/blob/master/COPYING
atomic	https://github.com/uber-go/atomic	https://github.com/uber-go/atomic/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
atomicwrites	https://github.com/untitaker/python-atomicwrites	https://github.com/untitaker/python-atomicwrites/blob/master/LICENSE
attrs	https://www.attrs.org/	https://github.com/python-attrs/attrs/blob/main/LICENSE
aws-sdk-go	https://github.com/aws/aws-sdk-go	https://github.com/aws/aws-sdk-go/blob/master/LICENSE.txt
babel	https://babel.pocoo.org/ OR https://github.com/python-babel/babel	https://github.com/python-babel/babel/blob/master/LICENSE OR https://babel.pocoo.org/en/latest/license.html#babel-license
Babel	https://github.com/python-babel/babel	https://github.com/python-babel/babel/blob/master/LICENSE
backcall	https://github.com/takluyver/backcall	https://github.com/takluyver/backcall/blob/master/LICENSE
backports	https://github.com/brandon-rhodes/backports	https://github.com/brandon-rhodes/backports/blob/master/LICENSE.TXT
backports.functools-lru-cache	https://github.com/jaraco/backports.functools_lru_cache	https://github.com/jaraco/backports.functools_lru_cache/blob/main/LICENSE
backports.zstd	https://github.com/rogdham/backports.zstd	https://github.com/rogdham/backports.zstd?tab=License-1-ov-file#readme
base62	https://github.com/kare/base62	https://github.com/kare/base62/blob/main/LICENSE
beautifulsoup4	https://www.crummy.com/software/BeautifulSoup/	https://www.crummy.com/software/BeautifulSoup/

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
billiard	https://github.com/celery/billiard	https://github.com/celery/billiard/blob/master/LICENSE.txt
biopython	https://biopython.org	https://github.com/biopython/biopython/blob/master/LICENSE.rst
biotite	https://github.com/biotite-dev/biotite/blob/master/README.rst	https://github.com/biotite-dev/biotite/blob/master/LICENSE.rst
bitarray	https://github.com/ilanschnell/bitarray	https://github.com/ilanschnell/bitarray/blob/master/LICENSE
bitset	https://github.com/willf/bitset	https://github.com/willf/bitset/blob/master/LICENSE
blas	https://www.netlib.org/blas/	https://www.netlib.org/blas/#_licensing
bleach	https://github.com/mozilla/bleach	https://github.com/mozilla/bleach/blob/main/LICENSE
blessings	https://github.com/erikrose/blessings	https://github.com/erikrose/blessings/blob/master/LICENSE
blinker	https://pythonhosted.org/blinker/	https://github.com/pallets-eco/blinker/blob/master/LICENSE
bokeh	https://github.com/bokeh/bokeh	https://github.com/bokeh/bokeh/blob/branch-4.0/LICENSE.txt
blosc	https://github.com/Blosc/python-blosc	https://github.com/Blosc/python-blosc/blob/master/LICENSE.txt
blosc2	https://github.com/Blosc/python-blosc2	https://github.com/Blosc/c-blosc2/blob/main/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
boltons	https://github.com/mahmoud/boltons	https://github.com/mahmoud/boltons/blob/master/LICENSE
Boltz	https://boltz.bio	https://github.com/jwohlwend/boltz/blob/main/LICENSE
boost	https://www.boost.org/doc/libs/1_71_0/libs/python/doc/html/index.html	https://www.boost.org/LICENSE_1_0.txt
boost-cpp	https://www.boost.org/doc/libs/1_71_0/libs/python/doc/html/index.html	https://www.boost.org/LICENSE_1_0.txt
bootstrap-vue	https://github.com/bootstrap-vue/bootstrap-vue	https://github.com/bootstrap-vue/bootstrap-vue/raw/master/LICENSE
boto3	https://github.com/boto/boto3	https://github.com/boto/boto3/blob/develop/LICENSE
botocore	https://github.com/boto/botocore	https://github.com/boto/botocore/blob/develop/LICENSE.txt
Bottleneck	https://bottleneck.readthedocs.io/en/latest/index.html	https://bottleneck.readthedocs.io/en/latest/license.html
Brotli	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
brotli-bin	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
brotli-python	http://python-hyper.org/projects/brotlipy/en/latest/	https://github.com/python-hyper/brotlicffi/blob/main/LICENSE
brotlicffi	https://github.com/python-hyper/brotlicffi	https://github.com/python-hyper/brotlicffi?tab=MIT-1-ov-file#readme
brotlipy	https://github.com/python-hyper/brotlicffi	https://github.com/python-hyper/brotlicffi/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
bson	http://github.com/py-bson/bson	https://github.com/py-bson/bson/blob/master/LICENSE_APACHE
bytefmt	https://code.cloudfoundry.org/bytefmt	https://github.com/cloudfoundry/bytefmt/blob/master/LICENSE
bzip2	https://www.bzip.org	https://spdx.org/licenses/bzip2-1.0.6.html
c-ares	https://github.com/c-ares/c-ares	https://github.com/c-ares/c-ares/blob/main/LICENSE.md
ca-certificates	https://github.com/conda-forge/ca-certificates-feedsto	https://wiki.mozilla.org/CA/Included_Certificates
cached-property	https://github.com/pydanny/cached-property	https://github.com/pydanny/cached-property/blob/master/LICENSE
cachetools	https://github.com/tkem/cachetools/	https://github.com/tkem/cachetools/blob/master/LICENSE
cairo	https://pycairo.readthedocs.io/en/latest/	https://github.com/pygobject/pycairo/blob/master/COPYING
canvas	https://github.com/Automattic/node-canvas	https://github.com/Automattic/node-canvas
cctbx	https://github.com/cctbx/cctbx_project	https://github.com/cctbx/cctbx_project/blob/master/cctbx/LICENSE_2_0.txt
celery	https://github.com/celery/celery	https://github.com/celery/celery/blob/master/LICENSE
Cerberus	https://docs.python-cerberus.org/en/stable/	https://docs.python-cerberus.org/en/stable/license.html
certifi	https://certifiio.readthedocs.io/en/latest/	https://wiki.mozilla.org/index.php?title=CA/IncludedCertificates&redirect=no

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
cffi	https://github.com/python-cffi	https://github.com/cffi/cffi/blob/master/COPYRIGHT
cftime	https://pypi.org/project/cftime/	https://github.com/Unidata/cftime/blob/master/LICENSE
chardet	https://github.com/chardet/chardet	https://github.com/chardet/chardet/blob/master/LICENSE
charset-normalizer	https://github.com/ousret/charset_normalizer	https://github.com/Ousret/charset_normalizer/blob/master/LICENSE
ChEMBL Structure Pipeline	https://github.com/chembl/ChEMBL_Structure_Pipeline/tree/master	https://github.com/chembl/ChEMBL_Structure_Pipeline/blob/master/LICENSE
choreographer	https://github.com/mazpie/choreographer	https://github.com/mazpie/choreographer/blob/main/LICENSE
chunkreader	https://github.com/jackc/chunkreader/v2	https://github.com/jackc/chunkreader/blob/master/LICENSE
click	https://palletsprojects.com/p/click/	https://github.com/pallets/click/blob/main/LICENSE.rst
click-completion	https://github.com/click-contrib/click-completion	https://github.com/click-contrib/click-completion/blob/master/LICENSE
click-didyoumean	https://github.com/click-contrib/click-didyoumean	https://github.com/click-contrib/click-didyoumean/blob/master/LICENSE
click-plugins	https://github.com/click-contrib/click-plugins	https://github.com/click-contrib/click-plugins/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
click-repl	https://github.com/untitaker/click-repl	https://github.com/click-contrib/click-repl/blob/master/LICENSE
cloudpickle	https://github.com/cloudpipe/cloudpickle	https://github.com/cloudpipe/cloudpickle/blob/master/LICENSE
cmake	https://cmake.org/	https://cmake.org/licensing/
colorama	https://github.com/tartley/colorama	https://github.com/tartley/colorama/blob/master/LICENSE.txt
comm	https://github.com/ipython/comm	https://github.com/ipython/comm/blob/master/LICENSE
compose	https://github.com/docker/compose	https://github.com/docker/compose/blob/v2/LICENSE
conda-content-trust	https://github.com/conda/conda-content-trust	https://github.com/conda/conda-content-trust/blob/main/LICENSE
conda-libmamba-solver	https://github.com/conda/conda-libmamba-solv	https://github.com/conda/conda-libmamba-solv/blob/main/LICENSE
conda-pack	https://conda.github.io/conda-pack/	https://github.com/conda/conda-pack/blob/main/LICENSE.txt
conda-package-handling	https://github.com/conda/conda-package-handli	https://github.com/conda/conda-package-handli/blob/main/LICENSE
conda-package-streaming	https://anaconda.org/conda-forge/conda-package-stream	https://github.com/conda/conda-package-stream/blob/main/LICENSE
conda-token	https://anaconda.org/anaconda/conda-token	https://anaconda.org/anaconda/conda-token

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
conda_package_strear	https://github.com/conda/conda-package-streamtree/main/conda_package_streaming	https://github.com/conda/conda-package-streamtab=License-1-ov-file
confuse	https://github.com/beetbox/confuse	https://github.com/beetbox/confuse/blob/main/LICENSE
contourpy	https://github.com/contourpy/contourpy	https://github.com/contourpy/contourpy/blob/main/LICENSE
cpp-expected	https://github.com/TartanLlama/expected	https://github.com/TartanLlama/expected?tab=CC0-1.0-1-ov-file#readme
cpp-peglib	https://github.com/yhirose/cpp-peglib	https://github.com/yhirose/cpp-peglib/blob/master/LICENSE
cryptography	https://github.com/pyca/cryptography	https://github.com/pyca/cryptography/blob/main/LICENSE
cssselect2	https://github.com/Kozea/cssselect2/	https://github.com/Kozea/cssselect2/blob/master/LICENSE
cuda-bindings	https://nvidia.github.io/cuda-python/cuda-bindings/latest/install.html	https://github.com/NVIDIA/cuda-python/blob/main/cuda_bindings/LICENSE
cuda-pathfinder	https://nvidia.github.io/cuda-python/cuda-pathfinder/latest/	https://nvidia.github.io/cuda-python/cuda-pathfinder/latest/license.html
cudatoolkit	https://developer.nvidia.com/cuda-toolkit	https://docs.nvidia.com/cuda/eula/index.html
cuequivariance-ops-cu12	https://developer.nvidia.com/cuequivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
cuequivariance-ops-torch-cu12	https://developer.nvidia.com/cuequivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
cuequivariance-torch	https://docs.nvidia.com/cuda/cuequivariance/	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
cupy-cuda113	https://cupy.dev/	https://github.com/cupy/cupy/blob/main/LICENSE
cupy-cuda11x	https://cupy.dev/	https://github.com/cupy/cupy/?tab=MIT-1-ov-file#readme
curl	https://curl.se	https://curl.se/docs/copyright.html
cycler	https://github.com/matplotlib/cycler	https://github.com/matplotlib/cycler/blob/main/LICENSE
cyrus-sasl	https://github.com/cyrusimap/cyrus-sasl	https://github.com/cyrusimap/cyrus-sasl/blob/master/COPYING
Cython	https://cython.org	https://www.apache.org/licenses/LICENSE-2.0.html
d3	https://github.com/mbostock/d3	https://github.com/mbostock/d3/raw/master/LICENSE
dataclasses	https://github.com/ericvsmith/dataclasses	https://github.com/ericvsmith/dataclasses/blob/master/LICENSE.txt
ddsketch	http://github.com/datadog/sketches-py	https://github.com/DataDog/sketches-py/blob/master/LICENSE
debugpy	https://aka.ms/debugpy	https://github.com/microsoft/debugpy/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
decimal	https://github.com/shopspring/decimal	https://github.com/shopspring/decimal/blob/master/LICENSE
decorator	https://github.com/micheles/decorator	https://github.com/micheles/decorator/blob/master/LICENSE.txt
deepdiff	https://github.com/seperman/deepdiff	https://github.com/seperman/deepdiff/blob/master/LICENSE
deeptime	https://github.com/deeptime-ml/deeptime	https://github.com/deeptime-ml/deeptime/blob/main/LICENSE.txt
defusedxml	https://github.com/tiran/defusedxml	https://github.com/tiran/defusedxml/blob/main/LICENSE
dftd3	https://github.com/dftd3/tad-dftd3	https://github.com/dftd3/tad-dftd3/blob/main/LICENSE
dill	https://github.com/uqfoundation/dill	https://github.com/uqfoundation/dill/blob/master/LICENSE
distlib	https://distlib.readthedocs.io/en/latest/	https://github.com/pypa/distlib/blob/master/LICENSE.txt
distro	https://github.com/python-distro/distro	https://github.com/python-distro/distro/blob/master/LICENSE
Django	https://www.djangoproject.com/	https://github.com/django/django/blob/main/LICENSE
django-classy-tags	https://github.com/django-cms/django-classy-tags	https://github.com/django-cms/django-classy-tags/blob/master/LICENSE
django-cors-headers	https://github.com/adamchainz/django-cors-headers	https://github.com/adamchainz/django-cors-headers/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
django-csp	https://github.com/mozilla/django-csp	https://github.com/mozilla/django-csp/blob/main/LICENSE
django-extensions	https://github.com/django-extensions/django-extensions	https://github.com/django-extensions/django-extensions/blob/main/LICENSE
django-filter	https://github.com/carltongibson/django-filter/tree/master	https://github.com/carltongibson/django-filter/blob/main/LICENSE
django-redis	https://github.com/jazzband/django-redis	https://github.com/jazzband/django-redis/blob/master/LICENSE
django-taggit	https://github.com/jazzband/django-taggit	https://github.com/jazzband/django-taggit/blob/master/LICENSE
django-taggit-serializer	https://github.com/glemmaPaul/django-taggit-serializ	https://github.com/glemmaPaul/django-taggit-serializ blob/master/LICENSE
django-taggit-templatetags2	https://github.com/fizista/django-taggit-templat	https://github.com/fizista/django-taggit-templat blob/master/LICENSE.txt
djangorestframework	https://www.django-rest-framewor org/	https://github.com/encode/django-rest-framewor blob/master/LICENSE.md
dkh	https://psicode.org/psi4manual/master/dkh.html	https://github.com/psi4/dkh/blob/master/COPYING.LESSER
dm-tree	https://github.com/deepmind/tree	https://github.com/deepmind/tree/blob/master/LICENSE
docker-py	https://github.com/docker/docker-py/	https://github.com/docker/docker-py/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
docker-pycreds	https://github.com/shin-/dockerpy-creds	https://github.com/shin-/dockerpy-creds/blob/master/LICENSE
docopt	https://docopt.org	https://github.com/docopt/docopt/blob/master/LICENSE-MIT
docutils	https://docutils.sourceforge.io	https://docutils.sourceforge.io/COPYING.html
drf-dynamic-fields	https://github.com/dbrgn/drf-dynamic-fields	https://github.com/dbrgn/drf-dynamic-fields/blob/master/LICENSE
DuckDB	https://duckdb.org/	https://github.com/duckdb/duckdb?tab=MIT-1-ov-file#readme
edgembar	https://rutgerslbsr.gitlab.io/fe-toolkit/edgembar/index.html	https://gitlab.com/RutgersLBSR/fe-toolkit/-/blob/master/LICENSE
editdistance	https://github.com/roy-ht/editdistance	https://github.com/roy-ht/editdistance/blob/master/LICENSE
eigen	https://eigen.tuxfamily.org/index.php?title=Main_Page	https://eigen.tuxfamily.org/index.php?title=Main_Page#License
einops	https://github.com/arogozhnikov/einops	https://github.com/arogozhnikov/einops/blob/master/LICENSE
einx	https://github.com/fferflo/einx	https://github.com/fferflo/einx/blob/master/LICENSE
entrypoints	https://github.com/takluyver/entrypoints	https://github.com/takluyver/entrypoints/blob/master/LICENSE
errors	https://github.com/pkg/errors	https://github.com/pkg/errors/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
eslint-plugin	https://github.com/typescript-eslint/typescript-eslint	https://github.com/typescript-eslint/typescript-eslint/raw/master/LICENSE
et_xmlfile	https://foss.heptapod.net/openpyxl/et_xmlfile	https://foss.heptapod.net/openpyxl/et_xmlfile/-/blob/branch/default/LICENCE.rst
exceptiongroup	https://github.com/agronholm/exceptiongroup	https://github.com/agronholm/exceptiongroup/blob/main/LICENSE
execnet	https://execnet.readthedocs.io/en/latest/	https://github.com/pytest-dev/execnet/blob/master/LICENSE
executing	https://github.com/alexmojaki/executing	https://github.com/alexmojaki/executing/blob/master/LICENSE.txt
expat	https://libexpat.github.io	https://github.com/libexpat/libexpat/blob/master/expat/COPYING
fairscale	https://github.com/facebookresearch/fairscale	https://github.com/facebookresearch/fairscale/blob/main/LICENSE
fairseq	https://github.com/pytorch/fairseq	https://github.com/facebookresearch/fairseq/blob/main/LICENSE
fastjsonschema	https://github.com/horejsek/python-fastjsonschem	https://github.com/horejsek/python-fastjsonschem/blob/master/LICENSE
fastrlock	https://github.com/scoder/fastrlock	https://github.com/scoder/fastrlock/blob/master/LICENSE
fftw	https://www.fftw.org	commercial license

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
filebuffer	https://github.com/mattetti/filebuffer	https://github.com/mattetti/filebuffer/blob/master/LICENSE
filelock	https://py-filelock.readthedocs.io/en/latest/index.html	https://github.com/tox-dev/py-filelock/blob/main/LICENSE
finufft	https://github.com/flatironinstitute/finufft	https://github.com/flatironinstitute/finufft/blob/master/LICENSE
Flask	https://flask.palletsprojects.com/en/2.1.x/	https://flask.palletsprojects.com/en/2.1.x/license/
flatbuffers	https://google.github.io/flatbuffers/	https://github.com/google/flatbuffers/blob/master/LICENSE.txt
flexcache	https://github.com/hgrecco/flexcache	https://github.com/hgrecco/flexcache/blob/main/LICENSE
flexparser	https://github.com/hgrecco/flexparser	https://github.com/hgrecco/flexparser/blob/main/LICENSE
flit-core	https://github.com/pypa/flit	https://github.com/scientific-python/lazy_loader/blob/main/LICENSE.md
FLTK	https://www.fltk.org/index.php	https://www.fltk.org/COPYING.php
fmt	https://fmt.dev/latest/index.html	https://github.com/fmtlib/fmt/blob/master/LICENSE.rst
font-awesome	https://github.com/FortAwesome/Font-Awesome	https://github.com/FortAwesome/Font-Awesome
font-ttf-dejavu-sans-mono	https://dejavu-fonts.github.io	https://dejavu-fonts.github.io/License.html
font-ttf-inconsolata	https://fonts.google.com/specimen/Inconsolata	https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
font-ttf-source-code-pro	https://fonts.google.com/specimen/Source+Code+Pro	https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL
font-ttf-ubuntu	https://fonts.google.com/specimen/Ubuntu	https://ubuntu.com/legal/font-licence
fontconfig	https://www.freedesktop.org/wiki/Software/fontconfig/	https://gitlab.freedesktop.org/fontconfig/fontconfig/-/blob/main/COPYING
fonts-conda-ecosystem	https://anaconda.org/conda-forge/fonts-conda-ecosyster	https://docs.conda.io/en/latest/license.html
fonts-conda-forge	https://anaconda.org/conda-forge/fonts-conda-forge/	https://creativecommons.org/licenses/by/4.0/
fonttools	https://github.com/fonttools/fonttools	https://github.com/fonttools/fonttools/blob/main/LICENSE
forcebalance	https://simtk.org/home/forcebalance OR https://github.com/leeping/forcebalance	https://github.com/leeping/forcebalance/blob/master/LICENSE
fqdn	https://github.com/ypcrts/fqdn	https://github.com/ypcrts/fqdn?tab=MPL-2.0-1-ov-file#readme
freetype	https://freetype.org	https://freetype.org/license.html
freetype-py	https://github.com/rougier/freetype-py	https://github.com/rougier/freetype-py?tab=License-1-ov-file#readme
fribidi	https://github.com/fribidi/fribidi	https://github.com/fribidi/fribidi/blob/master/COPYING
frozendict	https://github.com/Marco-Sulla/python-frozendict	https://github.com/Marco-Sulla/python-frozendict/blob/master/LICENSE.txt
frozenlist	https://github.com/aio-libs/frozenlist	https://github.com/aio-libs/frozenlist/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
fsmlite	https://github.com/tkem/fsmlite	https://github.com/tkem/fsmlite/blob/master/LICENSE
fsspec	https://github.com/fsspec/filesystem_spec	https://github.com/fsspec/filesystem_spec/blob/master/LICENSE
funcy	https://github.com/Suor/funcy	https://github.com/Suor/funcy/blob/master/LICENSE
gast	https://github.com/serge-sans-paille/gast/	https://github.com/serge-sans-paille/gast/blob/master/LICENSE
gau2grid	https://github.com/dgasmith/gau2grid	https://github.com/dgasmith/gau2grid/blob/master/LICENSE
gax-go	https://github.com/googleapis/gax-go/v2	https://github.com/googleapis/gax-go/blob/main/LICENSE
gdk-pixbuf	https://gitlab.gnome.org/GNOME/gdk-pixbuf	https://gitlab.gnome.org/GNOME/gdk-pixbuf/-/blob/master/COPYING
gemmi	https://gemmi.readthedocs.io/en/latest/	https://github.com/project-gemmi/gemmi/blob/master/LICENSE.txt
genproto	https://google.golang.org/genproto/googleapis	https://github.com/googleapis/go-genproto/blob/main/LICENSE
geometric	https://openbase.com/python/geometric	https://openbase.com/python/geometric
giflib	https://giflib.sourceforge.net	https://github.com/mirrorer/giflib/blob/master/COPYING
gitdb	https://github.com/gitpython-developers/gitdb	https://github.com/gitpython-developers/gitdb/blob/master/LICENSE
GitPython	https://github.com/gitpython-developers/GitPython	https://github.com/gitpython-developers/GitPython/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
glib	https://docs.gtk.org/glib/	https://gitlab.gnome.org/GNOME/glib/-/blob/main/LICENSES/LGPL-2.1-or-later.txt
GLM Library	https://github.com/g-truc/glm	https://github.com/g-truc/glm/blob/master/manual.md#section0
gls	https://github.com/jtolds/gls	https://github.com/jtolds/gls/blob/master/LICENSE
gmpy2	https://github.com/aleaxit/gmpy	https://github.com/gmpy2/gmpy2/blob/master/COPYING.LESSER
go-cleanhttp	https://github.com/hashicorp/go-cleanhttp	https://github.com/hashicorp/go-cleanhttp/blob/master/LICENSE
go-connections	https://github.com/docker/go-connections	https://github.com/docker/go-connections/blob/master/LICENSE
go-cpio	https://github.com/cavaliercoder/go-cpio	https://github.com/cavaliercoder/go-cpio/blob/master/LICENSE
go-getter	https://github.com/hashicorp/go-getter	https://github.com/hashicorp/go-getter/blob/master/LICENSE
go-homedir	https://github.com/mitchellh/go-homedir	https://github.com/mitchellh/go-homedir/blob/master/LICENSE
go-ini	https://github.com/go-ini/ini	https://github.com/go-ini/ini/blob/master/LICENSE
go-jmespath	https://github.com/jmespath/go-jmespath	https://github.com/jmespath/go-jmespath/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
go-junit-report	https://github.com/jstemmer/go-junit-report	https://github.com/jstemmer/go-junit-report/blob/master/LICENSE
go-netrc	https://github.com/bgentry/go-netrc/netrc	https://github.com/bgentry/go-netrc/blob/master/LICENSE
go-ole	https://github.com/go-ole/go-ole	https://github.com/go-ole/go-ole/blob/master/LICENSE
go-pg	https://github.com/go-pg/pg	https://github.com/go-pg/pg/blob/master/LICENSE
go-redis	https://github.com/go-redis/redis/v8	https://github.com/go-redis/redis/blob/master/LICENSE
go-rendezvous	https://github.com/dgryski/go-rendezvous	https://github.com/dgryski/go-rendezvous/blob/master/LICENSE
go-safetemp	https://github.com/hashicorp/go-safetemp	https://github.com/hashicorp/go-safetemp/blob/master/LICENSE
go-sysconf	https://github.com/tklauser/go-sysconf	https://github.com/tklauser/go-sysconf/blob/master/LICENSE
go-testing-interface	https://github.com/mitchellh/go-testing-interface	https://github.com/mitchellh/go-testing-interface/blob/master/LICENSE
go-units	https://github.com/docker/go-units	https://github.com/docker/go-units/blob/master/LICENSE
go-version	https://github.com/hashicorp/go-version	https://github.com/hashicorp/go-version/blob/master/LICENSE
go-zglob	https://github.com/mattn/go-zglob	https://github.com/mattn/go-zglob/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
go.opencensus	https://go.opencensus.io	https://github.com/census-instrumentatio opencensus-go/blob/master/LICENSE
gobrake.v2	https://gopkg.in/airbrake/gobrake.v2	https://github.com/airbrake/gobrake/blob/v2.0.9/LICENSE
goconvey	https://github.com/smartystreets/goconvey	https://github.com/smartystreets/goconvey/blob/master/LICENSE.md
golden-layout	https://github.com/deepstreamIO/golden-layout	https://github.com/deepstreamIO/golden-layout/raw/master/LICENSE
google-auth	https://google-auth.readthedocs.io/en/master/	https://github.com/GoogleCloudPlatform google-auth-library-p blob/master/LICENSE
google-auth-oauthlib	https://github.com/googleapis/google-auth-library-p	https://github.com/googleapis/google-auth-library-p blob/main/LICENSE
google-cloud-go	https://cloud.google.com/go	https://github.com/googleapis/google-cloud-go/blob/main/LICENSE
google-cloud-go/storage	https://cloud.google.com/go/storage	https://github.com/googleapis/google-cloud-go/blob/main/LICENSE
google-pasta	https://github.com/google/pasta	https://github.com/google/pasta/blob/master/LICENSE
google.golang.org/api	https://google.golang.org/api	https://github.com/googleapis/google-api-go-client/blob/main/LICENSE
gopsutil	https://github.com/shirou/gopsutil	https://github.com/shirou/gopsutil/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
gorilla websocket	https://github.com/gorilla/websocket	https://github.com/gorilla/websocket/blob/master/LICENSE
gpytorch	https://gpytorch.ai	https://github.com/cornellius-gp/gpytorch?tab=MIT-1-ov-file#readme
graphite2	https://github.com/silnrsi/graphite	https://github.com/silnrsi/graphite/blob/master/LICENSE
graphviz	https://graphviz.org/	https://graphviz.org/license/
greenlet	https://greenlet.readthedocs.io/en/latest/	https://github.com/python-greenlet/greenlet/blob/master/LICENSE
groupcache	https://github.com/golang/groupcache	https://github.com/golang/groupcache/blob/master/LICENSE
grpc	https://google.golang.org/grpc	https://github.com/grpc/grpc-go/blob/master/LICENSE
grpc-cpp	https://github.com/grpc/grpc	https://github.com/grpc/grpc/blob/master/LICENSE
grpcio	https://github.com/grpc/grpc.io/blob/main/LICENSE	https://github.com/grpc/grpc.io/blob/main/LICENSE
gtk2	https://gitlab.gnome.org/GNOME/gtk	https://gitlab.gnome.org/GNOME/gtk/-/blob/main/COPYING
gts	https://sourceforge.net/projects/gts/	https://www.gnu.org/licenses/lgpl-3.0.html
h11	https://github.com/python-hyper/h11	https://github.com/python-hyper/h11/blob/master/LICENSE.txt
h2	https://github.com/python-hyper/h2	https://github.com/python-hyper/h2?tab=MIT-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
h5py	https://www.h5py.org	https://docs.h5py.org/en/stable/licenses.html
harfbuzz	https://github.com/harfbuzz/harfbuzz	https://github.com/harfbuzz/harfbuzz/blob/main/COPYING
hdbscan	https://hdbscan.readthedocs.io/en/latest/	https://github.com/scikit-learn-contrib/hdbscan/blob/master/LICENSE
hdf4	https://www.hdfgroup.org/solutions/hdf4/	https://github.com/HDFGroup/hdf4/blob/master/COPYING
hdf5	https://www.hdfgroup.org/solutions/hdf5/	https://github.com/HDFGroup/hdf5/blob/master/COPYING
he	https://github.com/mathiasbynens/he	https://github.com/mathiasbynens/he/raw/master/LICENSE-MIT.txt
hpack	https://github.com/python-hyper/hpack	https://github.com/python-hyper/hpack?tab=MIT-1-ov-file#readme
html-loader	https://github.com/webpack-contrib/html-loader	https://github.com/webpack-contrib/html-loader/raw/master/LICENSE
html5lib	https://github.com/html5lib/html5lib-python	https://github.com/html5lib/html5lib-python/blob/master/LICENSE
htslib	https://www.htslib.org	https://github.com/samtools/htslib/blob/develop/LICENSE
httpcore	https://www.encode.io/httpcore/	https://github.com/encode/httpcore/blob/master/LICENSE.md
httpx	https://github.com/encode/httpx	https://github.com/encode/httpx/blob/master/LICENSE.md

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
humanize	https://github.com/jmoiron/humanize	https://github.com/jmoiron/humanize/blob/master/LICENCE
hydra-core	https://github.com/facebookresearch/hydra	https://github.com/facebookresearch/hydra/blob/main/LICENSE
hyperframe	https://github.com/python-hyper/hyperframe	https://github.com/python-hyper/hyperframe?tab=MIT-1-ov-file#readme
icu	https://github.com/unicode-org/icu	https://github.com/unicode-org/icu/blob/main/icu4c/LICENSE
idna	https://github.com/kjd/idna	https://github.com/kjd/idna/blob/master/LICENSE.md
ihm	https://github.com/ihmwg/python-ihm	https://github.com/ihmwg/python-ihm/blob/main/LICENSE
imageio	https://github.com/imageio/imageio	https://github.com/imageio/imageio/blob/master/LICENSE
imagesize	https://github.com/shibukawa/imagesize_py	https://github.com/shibukawa/imagesize_py/blob/master/LICENSE.rst
ImmuneBuilder	https://github.com/oxpig/ImmuneBuilder/tree/main	https://github.com/oxpig/ImmuneBuilder/blob/main/LICENSE
importlib-metadata	https://github.com/python/importlib_metadata	https://github.com/python/importlib_metadata/blob/main/LICENSE
importlib_metadata	https://github.com/python/importlib_metadata	https://github.com/python/importlib_metadata/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
importlib_resources	https://github.com/python/importlib_resources	https://github.com/python/importlib_resources/blob/main/LICENSE
InChI	https://iupac.org/who-we-are/divisions/division-details/inchi/	https://iupac.org/wp-content/uploads/2021/06/iupac-inchi-license_2020.pdf
inflection	https://github.com/jinzhu/inflection	https://github.com/jinzhu/inflection/blob/master/LICENSE
ini.v1	https://gopkg.in/ini.v1	https://github.com/go-ini/ini/blob/v1.66.2/LICENSE
iniconfig	https://github.com/pytest-dev/iniconfig	https://github.com/pytest-dev/iniconfig/blob/main/LICENSE
innersvg-polyfill	https://github.com/dnozay/innersvg-polyfill	https://github.com/dnozay/innersvg-polyfill/raw/master/LICENSE
intel-openmp	https://github.com/hermitcore/libomp_oss	https://github.com/hermitcore/libomp_oss/blob/hermit/LICENSE.txt
ipykernel	https://ipython.org	https://ipython.org
ipython	https://ipython.org	https://github.com/ipython/ipython/blob/master/LICENSE
ipython-genutils	http://ipython.org	https://github.com/ipython/ipython_genutils/blob/master/COPYING.md
ipython_pygments_le	https://github.com/ipython/ipython-pygments-lex	https://github.com/ipython/ipython-pygments-lex blob/main/LICENSE
ipywidgets	http://jupyter.org	https://github.com/jupyter-widgets/ipywidgets/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
isodate	https://github.com/gweis/isodate/	https://github.com/gweis/isodate/blob/master/LICENSE
isoduration	https://github.com/bolsote/isoduration	https://github.com/bolsote/isoduration?tab=ISC-1-ov-file#readme
itsdangerous	https://palletsprojects.com/p/itsdangerous/	https://itsdangerous.palletsprojects.com/en/2.1.x/license/
jax	https://github.com/google/jax	https://github.com/google/jax/blob/main/LICENSE
jaxlib	https://github.com/google/jax	https://github.com/google/jax/blob/main/LICENSE
jaxtyping	https://github.com/patrick-kidger/jaxtyping	https://github.com/patrick-kidger/jaxtyping?tab=License-1-ov-file#readme
jedi	https://jedi.readthedocs.io/en/latest/	https://github.com/davidhalter/jedi/blob/master/LICENSE.txt
Jinja2	https://palletsprojects.com/p/jinja/	https://github.com/pallets/jinja/blob/main/LICENSE.rst
jmespath	https://github.com/jmespath/jmespath.py	https://github.com/jmespath/jmespath.py/blob/develop/LICENSE.txt
joblib	https://joblib.readthedocs.io/en/latest/	https://github.com/joblib/joblib/blob/master/LICENSE.txt
jpeg	https://www.ijg.org	https://www.ijg.org/files/README
json5	https://github.com/dpranke/pyjson5	https://github.com/dpranke/pyjson5/blob/main/LICENSE
jsonfield	https://github.com/dmkoch/django-jsonfield/	https://github.com/rpkilby/jsonfield/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
jsonpatch	https://github.com/stefankoegl/python-json-patch	https://github.com/stefankoegl/python-json-patch/blob/master/LICENSE
jsonpickle	https://github.com/jsonpickle/jsonpickle	https://github.com/jsonpickle/jsonpickle/blob/main/LICENSE
jsonpointer	https://github.com/stefankoegl/python-json-pointer	https://github.com/stefankoegl/python-json-pointer/blob/master/LICENSE.txt
jsonschema	https://github.com/python-jsonschema/jsonschema	https://github.com/python-jsonschema/jsonschema/blob/master/COPYING
jsonschema-specifications	https://github.com/python-jsonschema/jsonschema-specificat	https://github.com/python-jsonschema/jsonschema-specificat blob/main/COPYING
jstat	https://github.com/jstat/jstat	https://www.opensource.org/licenses/mit-license.php
jupyter-client	https://jupyter.org	https://github.com/jupyter/jupyter_client/blob/main/COPYING.md
jupyter-core	https://jupyter.org	https://github.com/jupyter/jupyter_core/blob/master/COPYING.md
jupyter-events	https://github.com/jupyter/jupyter_events	https://github.com/jupyter/jupyter_events/blob/main/LICENSE
jupyter-lsp	https://github.com/jupyter-lsp/jupyterlab-lsp	https://github.com/jupyter-lsp/jupyterlab-lsp
jupyter-serverhttp://jupyter.or	https://github.com/jupyter-server/jupyter_server/blob/main/LICENSE	
jupyter_client	http://jupyter.org	https://github.com/jupyter/jupyter_client/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
jupyter_core	http://jupyter.org	https://github.com/jupyter/jupyter_core/blob/master/LICENSE
jupyter_server	https://github.com/jupyter-server/jupyter_server	https://github.com/jupyter-server/jupyter_server/blob/main/LICENSE
jupyter_server_termin	https://github.com/jupyter-server/jupyter_server_terminals	https://github.com/jupyter-server/jupyter_server_terminals/blob/main/LICENSE
jupyterlab	https://github.com/jupyterlab/jupyterlab	https://github.com/jupyterlab/jupyterlab/blob/main/LICENSE
jupyterlab-pygments	http://jupyter.org	https://github.com/jupyterlab/jupyterlab_pygments/blob/main/LICENSE
jupyterlab-widgets	http://jupyter.org	https://github.com/jupyter-widgets/ipywidgets/blob/master/LICENSE
jupyter-lab_pygments	https://github.com/jupyterlab/jupyterlab_pygments	https://github.com/jupyterlab/jupyterlab_pygments?tab=BSD-3-Clause-1-ov-file#readme
jupyterlab_server	https://github.com/jupyterlab/jupyterlab_server	https://github.com/jupyterlab/jupyterlab_server/blob/main/LICENSE
jupyterlab_widgets	https://github.com/jupyter-widgets/ipywidgets	https://github.com/jupyter-widgets/ipywidgets?tab=BSD-3-Clause-1-ov-file#readme
kaleido	https://github.com/plotly/Kaleido	https://github.com/plotly/Kaleido/blob/master/LICENSE.txt
keras	https://github.com/keras-team/keras	https://github.com/keras-team/keras/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
Keras-Preprocessing	https://github.com/keras-team/keras-preprocessing	https://github.com/keras-team/keras-preprocessing/blob/master/LICENSE
keras-tuner	https://github.com/keras-team/keras-tuner	https://github.com/keras-team/keras-tuner/blob/master/LICENSE
keyring	https://github.com/jaraco/keyring	https://github.com/jaraco/keyring/blob/main/LICENSE
keyutils	https://github.com/sassoftware/python-keyutils	https://github.com/sassoftware/python-keyutils/blob/master/LICENSE
kiwisolver	https://kiwisolver.readthedocs.io/en/latest/	https://github.com/nucleic/kiwi/blob/main/LICENSE
kombu	https://kombu.readthedocs.io	https://github.com/celery/kombu/blob/master/LICENSE
krb5	https://web.mit.edu/kerberos/	https://github.com/krb5/krb5/blob/master/src/lib/gssapi/LICENSE
kt-legacy	https://github.com/haifeng-jin/kt-legacy	https://github.com/haifeng-jin/kt-legacy/blob/main/LICENSE
lark	https://github.com/lark-parser/lark	https://github.com/lark-parser/lark?tab=MIT-1-ov-file#readme
lazy_loader	https://github.com/scientific-python/lazy_loader	https://github.com/scientific-python/lazy_loader/blob/main/LICENSE.md
lcms2	https://www.littlecms.com	https://github.com/sk1project/python-lcms2/blob/master/LICENSE
legacy-cgi	https://github.com/jackrosenthal/legacy-cgi	https://github.com/jackrosenthal/legacy-cgi?tab=License-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
lerc	https://github.com/Esri/lerc	https://github.com/Esri/lerc/blob/master/LICENSE
libarchive	http://www.libarchive.org	https://raw.githubusercontent.com/libarchive/libarchive/master/COPYING
libblas	https://www.netlib.org/blas/	https://www.netlib.org/blas/#_licensing
libbrotlicommon	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libbrotlidec	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libbrotlienc	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libcblas	https://anaconda.org/conda-forge/libcblas	N/A
libclang	https://github.com/sighingnow/libclang	https://github.com/sighingnow/libclang/blob/master/LICENSE.TXT
libcurl	https://curl.se/libcurl/	https://curl.se/docs/copyright.html
libcxx	https://github.com/llvm-mirror/libcxx	https://github.com/llvm-mirror/libcxx/blob/master/LICENSE.TXT
libdb	https://www.oracle.com/technology/software/products/berkeley-db/index.html	https://github.com/berkeleydb/libdb/blob/master/LEGAL
libdeflate	https://github.com/ebiggers/libdeflate	https://github.com/ebiggers/libdeflate/blob/master/COPYING
libedit	https://thrysoee.dk/editline/	http://www.netbsd.org/Goals/redistribution.html
libev	https://software.schmorp.de/pkg/libev.html	https://software.schmorp.de/pkg/libev.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libexpat	https://libexpat.github.io/	https://github.com/libexpat/libexpat?tab=MIT-1-ov-file#readme
libffi	https://github.com/libffi/libffi	https://github.com/libffi/libffi/blob/master/LICENSE
libgcc	https://gcc.gnu.org/onlinedocs/gccint/Libgcc.html	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
libgcrypt	https://gnupg.org/software/index.html	https://git.gnupg.org/cgi-bin/gitweb.cgi?p=libgcrypt.git;a=blob;f=COPYING.LIB;hb=HEAD
libgd	https://libgd.github.io	https://libgd.github.io/manuals/2.2.1/files/license-txt.html
libglib	https://github.com/PolMine/libglib	https://github.com/PolMine/libglib/blob/master/include/glib-2.0/glib.h
libiconv	https://www.gnu.org/software/libiconv/	https://www.gnu.org/software/libiconv/#downloading
libint	https://tinyurl.com/yvw97wbw	https://tinyurl.com/yvw97wbw
liblapack	http://www.netlib.org/lapack/	https://www.netlib.org/lapack/LICENSE.txt
liblapacke	https://anaconda.org/conda-forge/liblapacke	https://www.netlib.org/lapack/LICENSE.txt
liblzma	https://tukaani.org/xz/	https://git.tukaani.org/?p=xz.git;a=blob;f=COPYING.0BSD
libmamba	https://medium.com/@QuantStack/open-software-packag	https://github.com/mamba-org/mamba/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libmambapy	https://www.anaconda.com/blog/a-faster-conda-for-a-g	https://github.com/conda/conda-libmamba-solv blob/main/LICENSE
libnetcdf	https://www.unidata.ucar.edu/software/netcdf/	https://www.unidata.ucar.edu/software/netcdf/copyright.html
libnghttp2	https://github.com/nghttp2/nghttp2	https://github.com/nghttp2/nghttp2/blob/master/COPYING
libnsl	https://github.com/thkukuk/libnsl	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
libopenblas	https://www.openblas.net/	https://github.com/xianyi/OpenBLAS/blob/develop/LICENSE
libpng	https://www.libpng.org/pub/png/libpng.html	https://www.libpng.org/pub/png/src/libpng-LICENSE.txt
libpq	https://www.postgresql.org/	https://www.postgresql.org/about/licence/
librsvg	https://gitlab.gnome.org/GNOME/librsvg	https://gitlab.gnome.org/GNOME/librsvg/-/blob/main/COPYING.LIB
libsolv	https://github.com/openSUSE/libsolv	https://github.com/openSUSE/libsolv/blob/master/LICENSE.BSD
libsqlite	https://github.com/mackyle/sqlite	https://github.com/mackyle/sqlite?tab=License-1-ov-file#readme
libssh2	https://github.com/libssh2/libssh2	https://github.com/libssh2/libssh2/blob/master/COPYING
libstdcxx	https://gcc.gnu.org/	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libtiff	https://www.libtiff.org/	https://www.libtiff.org/misc.html
libtrust	https://github.com/docker/libtrust	https://github.com/docker/libtrust/blob/master/LICENSE
libuuid	https://sourceforge.net/projects/libuuid/	https://sourceforge.net/p/libuuid/code/ci/master/tree/COPYING
libuv	https://github.com/libuv/libuv	https://github.com/libuv/libuv/blob/v1.x/LICENSE
libwebp	https://chromium.googlesource.com/?format=HTML	https://chromium.googlesource.com/chromium/src/third_party/+/refs/heads/main/libwebp/LICENSE
libwebp-base	https://chromium.googlesource.com/?format=HTML	https://chromium.googlesource.com/chromium/src/third_party/+/refs/heads/main/libwebp/LICENSE
libxc	https://www.tddft.org/programs/libxc/	https://gitlab.com/libxc/libxc/-/blob/master/COPYING
libxcb	https://xcb.freedesktop.org	https://github.com/freedesktop/xcb-libxcb/blob/master/COPYING
libxcrypt	https://github.com/besser82/libxcrypt	https://github.com/besser82/libxcrypt?tab=LGPL-2.1-1-ov-file#readme
libxml2	https://git.gnome.org/browse/libxml2/	https://gitlab.gnome.org/GNOME/libxml2/-/blob/master/Copyright
libxmlsec1	https://github.com/lsh123/xmlsec	https://github.com/lsh123/xmlsec/blob/master/README
libxslt	https://gitlab.gnome.org/GNOME/libxslt/-/wikis/home	https://gitlab.gnome.org/GNOME/libxslt/-/blob/master/Copyright

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libzlib	https://zlib.net	https://zlib.net/zlib_license.html
lightning-utilities	https://github.com/Lightning-AI/utilities	https://github.com/Lightning-AI/utilities?tab=Apache-2.0-1-ov-file#readme
lime	https://github.com/marcotcr/lime	https://github.com/marcotcr/lime/blob/master/LICENSE
linear-operator	https://github.com/cornellius-gp/linear_operator	https://github.com/cornellius-gp/linear_operator?tab=MIT-1-ov-file#readme
lit	http://llvm.org	https://releases.llvm.org/14.0.0/LICENSE.TXT
llvm-openmp	https://github.com/llvm-mirror/openmp	https://github.com/llvm-mirror/openmp/blob/master/LICENSE.txt
llvmlite	http://llvmlite.readthedocs.io	https://github.com/numba/llvmlite/blob/main/LICENSE
loader-utils	https://github.com/webpack/loader-utils	https://github.com/webpack/loader-utils/raw/master/LICENSE
logistro	https://github.com/geopozo/logistro	https://github.com/geopozo/logistro?tab=MIT-1-ov-file#readme
logomaker	https://logomaker.readthedocs.io/en/latest/	https://github.com/jbkinney/logomaker/blob/master/LICENSE
logrus	https://github.com/sirupsen/logrus	https://github.com/sirupsen/logrus/blob/master/LICENSE
logrus-airbrake-hook.v2	https://gopkg.in/gemnasium/logrus-airbrake-hook.v2	https://github.com/gemnasium/logrus-airbrake-hook/blob/v2.1.2/LICENSE
lxml	https://lxml.de	https://lxml.de/index.html#license

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
lz4-c	https://www.lz4.org/	https://github.com/lz4/lz4/blob/dev/LICENSE
lzo	https://www.oberhumer.com/opensource/lzo/	https://www.oberhumer.com/opensource/gpl.html
mamba	https://github.com/mamba-org/mamba	https://github.com/mamba-org/mamba?tab=BSD-3-Clause-1-ov-f readme
markdown	https://github.com/evilstreak/markdown-js	https://www.opensource.org/licenses/mit-license.php
markdown-it-py	https://github.com/executablebooks/markdown-it-py	https://github.com/executablebooks/markdown-it-py/blob/master/LICENSE
markupsafe	https://github.com/pallets/markupsafe/tree/main/src/markupsafe	https://github.com/pallets/markupsafe/tree/main?tab=BSD-3-Clause-1-ov-f
MarkupSafe	https://palletsprojects.com/p/markupsafe/	https://github.com/pallets/markupsafe/blob/main/LICENSE.rst
mashumaro	https://github.com/Fatal1ty/mashumaro	https://github.com/Fatal1ty/mashumaro
matplotlib	https://matplotlib.org	https://matplotlib.org/stable/users/project/license.html
matplotlib-base	https://matplotlib.org	https://matplotlib.org/stable/users/project/license.html
matplotlib-inline	https://github.com/ipython/matplotlib-inline	https://github.com/ipython/matplotlib-inline/blob/master/LICENSE
mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
mda_xdrlib	https://github.com/MDAnalysis/mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib?tab=License-1-ov-file#readme
mdtraj	https://www.mdtraj.org/	https://www.mdtraj.org/1.9.8.dev0/index.html#license
mdurl	https://github.com/markdown-it/mdurl	https://github.com/markdown-it/mdurl/blob/master/LICENSE
menuinst	https://github.com/conda/menuinst	https://github.com/conda/menuinst/blob/main/LICENSE.txt
mergo	https://github.com/imdario/mergo	https://github.com/imdario/mergo/blob/master/LICENSE
mistune	https://github.com/lepture/mistune	https://github.com/lepture/mistune/blob/master/LICENSE
mkl	https://github.com/rust-math/intel-mkl-src	https://github.com/rust-math/intel-mkl-src/blob/master/License.txt
mkl-fft	https://github.com/IntelPython/mkl_fft	https://github.com/IntelPython/mkl_fft/blob/master/LICENSE.txt
mkl-random	https://github.com/IntelPython/mkl_random	https://github.com/IntelPython/mkl_random/blob/master/LICENSE.txt
mkl-service	https://github.com/IntelPython/mkl-service	https://github.com/IntelPython/mkl-service/blob/master/LICENSE.txt
mkl_fft	http://github.com/IntelPython/mkl_fft	https://github.com/IntelPython/mkl_fft?tab=BSD-3-Clause-1-ov-f readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
mkl_random	http://github.com/IntelPython/mkl_random	https://github.com/IntelPython/mkl_random?tab=BSD-3-Clause-1-ov-f readme
ml-dtypes	https://github.com/jax-ml/ml_dtypes	https://github.com/jax-ml/ml_dtypes/blob/main/LICENSE
ml_dtypes	https://github.com/jax-ml/ml_dtypes	https://github.com/jax-ml/ml_dtypes/blob/main/LICENSE
modelcif	https://github.com/ihmwg/python-modelcif	https://github.com/ihmwg/python-modelcif/blob/main/LICENSE
molecupy	https://molecupy.readthedocs.io/en/latest/	https://github.com/samirelanduk/atomium/blob/master/LICENSE
moment	https://github.com/moment/moment	https://github.com/moment/moment/raw/master/LICENSE
moment-precise-range-plugin	https://github.com/codebox/moment-precise-rang	https://github.com/codebox/moment-precise-rang raw/master/LICENSE.md
more-itertools	https://github.com/more-itertools/more-itertools	https://github.com/more-itertools/more-itertools/blob/master/LICENSE
mpiplus	https://github.com/choderalab/mpiplus	https://github.com/choderalab/mpiplus/blob/main/LICENSE
mpmath	http://mpmath.org/	https://github.com/mpmath/mpmath/blob/master/LICENSE
mrcfile	https://github.com/ccpem/mrcfile	https://github.com/ccpem/mrcfile/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
msgpack	https://msgpack.org/	https://github.com/msgpack/msgpack-python/blob/main/COPYING
multidict	https://github.com/aio-libs/multidict	https://github.com/aio-libs/multidict/blob/master/LICENSE
multierr	https://go.uber.org/multierr	https://github.com/uber-go/multierr/blob/master/LICENSE.txt
multiprocess	https://github.com/uqfoundation/multiprocess	https://github.com/uqfoundation/multiprocess/blob/master/LICENSE
munkres	https://software.clapper.org/munkres/	https://github.com/bmc/munkres/blob/master/LICENSE.md
myesui uuid	https://github.com/myesui/uuid	https://pkg.go.dev/github.com/myesui/uuid?tab=licenses
namex	https://github.com/bcgov/namex	https://github.com/bcgov/namex/blob/main/LICENSE
narwhals	https://github.com/narwhals-dev/narwhals	https://github.com/narwhals-dev/narwhals?tab=MIT-1-ov-file#readme
nbclassic	http://jupyter.org	https://github.com/jupyter/nbclassic/blob/main/LICENSE
nbclient	https://jupyter.org	https://github.com/jupyter/nbclient/blob/main/LICENSE
nbconvert	https://jupyter.org	https://github.com/jupyter/nbconvert/blob/main/LICENSE
nbformat	http://jupyter.org	https://github.com/jupyter/nbformat/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
ncurses	https://invisible-island.net/ncurses/	https://invisible-island.net/ncurses/ncurses-license.html
ndfes	https://rutgerslbsr.gitlab.io/fe-toolkit/	https://gitlab.com/RutgersLBSR/fe-toolkit/-/blob/master/LICENSE
ndindex	https://quansight-labs.github.io/ndindex/	https://github.com/Quansight-Labs/ndindex?tab=MIT-1-ov-file#readme
nest-asyncio	https://github.com/erdewit/nest_asyncio	https://github.com/erdewit/nest_asyncio/blob/master/LICENSE
nest_asyncio	https://github.com/erdewit/nest_asyncio	https://github.com/erdewit/nest_asyncio?tab=BSD-2-Clause-1-ov-f readme
netcdf-fortran	https://www.unidata.ucar.edu/software/netcdf/	https://www.unidata.ucar.edu/software/netcdf/copyright.html
netCDF4	http://github.com/Unidata/netcdf4-python	https://github.com/Unidata/netcdf4-python/blob/master/LICENSE
nettle	https://git.lysator.liu.se/nettle/nettle	https://git.lysator.liu.se/nettle/nettle/-/blob/master/AUTHORS?ref_type=heads
networkx	https://networkx.org	https://github.com/networkx/networkx/blob/main/LICENSE.txt
nfpm	https://github.com/goreleaser/nfpm	https://github.com/goreleaser/nfpm/blob/master/LICENSE.md
ng-tags-input	https://github.com/mbenford/ngTagsInput	https://github.com/mbenford/ngTagsInput
ng-toast	https://github.com/tameraydin/ngToast	https://github.com/tameraydin/ngToast

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
ngdraggable	https://github.com/fatlinesofcode/ngDraggable	https://github.com/fatlinesofcode/ngDraggable/raw/master/LICENSE.txt
ngVue	https://github.com/ngVue/ngVue	https://github.com/ngVue/ngVue/raw/master/LICENSE
nh3	https://nh3.readthedocs.io/en/latest/	https://github.com/messense/nh3/blob/main/LICENSE
nlohmann_json	https://github.com/nlohmann/json	https://github.com/nlohmann/json?tab=MIT-1-ov-file#readme
nlopt	https://nlopt.readthedocs.io/en/latest/NLopt_Python_Reference/	https://nlopt.readthedocs.io/en/latest/NLopt_License_and_Copyright/
nodejs	https://nodejs.org/en/	https://raw.githubusercontent.com/nodejs/node/master/LICENSE
nomkl	https://github.com/conda-forge/nomkl-feedstock	https://github.com/conda-forge/nomkl-feedstock/blob/main/LICENSE.txt
notebook	http://jupyter.org	https://github.com/jupyter/notebook/blob/main/LICENSE
notebook-shim	https://github.com/jupyter/notebook_shim	https://github.com/jupyter/notebook_shim
notebook_shim	http://jupyter.org	https://github.com/jupyter/notebook_shim/blob/main/LICENSE
numba	https://numba.pydata.org	https://github.com/numba/numba/blob/main/LICENSE
numcpus	https://github.com/tklauser/numcpus	https://github.com/tklauser/numcpus/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
numexpr	https://github.com/pydata/numexpr	https://github.com/pydata/numexpr/blob/master/LICENSE.txt
numpy	https://numpy.org	https://github.com/numpy/numpy/blob/main/LICENSE.txt
numpy-base	https://numpy.org	https://github.com/numpy/numpy/blob/main/LICENSE.txt
numpydoc	https://numpydoc.readthedocs.io/en/latest/index.html	https://github.com/numpy/numpydoc/blob/main/LICENSE.txt
nvidia-cublas-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cublas-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-cupti-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-cupti-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-nvrtc-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-nvrtc-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-runtime-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-runtime-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cudnn-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/deeplearning/cudnn/sla/index.html
nvidia-cudnn-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/deeplearning/cudnn/sla/index.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
nvidia-cuEquivariance	https://github.com/NVIDIA/cuEquivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
nvidia-cufft-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cufft-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cufile-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html#
nvidia-curand-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-curand-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusolver-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusolver-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparse-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparse-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparselt-cu12	https://docs.nvidia.com/cuda/cusparselt/index.html	https://docs.nvidia.com/cuda/cusparselt/license.html
nvidia-ml-py	https://pypi.org/project/nvidia-ml-py/	https://pypi.org/project/nvidia-ml-py/
nvidia-nccl-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/hpc-sdk/eula/index.html
nvidia-nccl-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/hpc-sdk/eula/index.html
nvidia-nvjitlink-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
nvidia-nvshmem-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-nvtx-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-nvtx-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
Oat++	https://oatpp.io/	https://github.com/oatpp/oatpp/blob/master/LICENSE
oauthlib	https://github.com/oauthlib/oauthlib	https://github.com/oauthlib/oauthlib/blob/master/LICENSE
ocl-icd	https://github.com/OCL-dev/ocl-icd	https://github.com/OCL-dev/ocl-icd/blob/master/COPYING
ocl-icd-system	https://github.com/conda-forge/ocl-icd-system-feedst	https://github.com/conda-forge/ocl-icd-system-feedst blob/main/LICENSE.txt
olefile	https://www.decalage.info/python/olefileio	https://olefile.readthedocs.io/en/latest/License.html
omegaconf	https://github.com/omry/omegaconf	https://github.com/omry/omegaconf/blob/master/LICENSE
OmegaFold	https://github.com/HeliXonProtein/OmegaFold/tree/main	https://github.com/HeliXonProtein/OmegaFold/blob/main/LICENSE
omnicanvas	https://omnicanvas.readthedocs.io/en/latest/	https://github.com/samirelanduk/omnicanvas/blob/master/LICENSE
OpenFF	https://openforcefield.org/	https://github.com/openforcefield/openff-forcefields
openff-amber-ff-ports	https://github.com/openforcefield/openff-amber-ff-ports	https://github.com/openforcefield/openff-amber-ff-ports blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
openff-forcefields	https://openforcefield.org	https://github.com/openforcefield/openff-forcefields/blob/master/LICENSE
openff-fragmenter	https://github.com/openforcefield/openff-fragmenter	https://github.com/openforcefield/openff-fragmenter/blob/main/LICENSE
openff-interchange	https://github.com/openforcefield/openff-interchange	https://github.com/openforcefield/openff-interchange/blob/main/LICENSE
openff-models	https://github.com/openforcefield/openff-models	https://github.com/openforcefield/openff-models/blob/main/LICENSE
openff-nagl	https://docs.openforcefield.org/projects/nagl/en/latest/?badge=latest	https://github.com/openforcefield/openff-nagl/blob/main/LICENSE
openff-nagl-models	https://github.com/openforcefield/openff-nagl-models	https://github.com/openforcefield/openff-nagl-models/blob/main/LICENSE
openff-toolkit	https://openforcefield.org	https://github.com/openforcefield/openff-toolkit/blob/master/LICENSE
openff-toolkit-base	https://openforcefield.org	https://github.com/openforcefield/openff-toolkit/blob/master/LICENSE
openff-units	https://github.com/openforcefield/openff-units	https://github.com/openforcefield/openff-units/blob/main/LICENSE
openff-utilities	https://github.com/openforcefield/openff-utilities	https://github.com/openforcefield/openff-utilities/blob/main/LICENSE
openjpeg	https://github.com/uclouvain/openjpeg	https://github.com/uclouvain/openjpeg/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
openldap	https://www.openldap.org/software/repo.html	https://git.openldap.org/openldap/openldap/-/blob/master/LICENSE
OpenMM	https://openmm.org	https://docs.openmm.org/7.1.0/userguide/library.html#license
openmmtools	https://github.com/choderalab/openmmtools	https://github.com/choderalab/openmmtools/blob/main/LICENSE
openmoltools	https://github.com/choderalab/openmoltools	https://github.com/choderalab/openmoltools/blob/master/LICENSE
openpyxl	https://openpyxl.readthedocs.io/en/stable/	https://foss.heptapod.net/openpyxl/openpyxl/-/blob/branch/3.0/LICENCE.rst
openssl	https://www.openssl.org	https://github.com/openssl/openssl/blob/master/LICENSE.txt
opt-einsum	https://github.com/dgasmith/opt_einsum	https://github.com/dgasmith/opt_einsum/blob/master/LICENSE
opt_einsum	https://github.com/dgasmith/opt_einsum	https://github.com/dgasmith/opt_einsum/blob/main/LICENSE
OptKing	https://github.com/psi-rking/optking	https://github.com/psi-rking/optking/blob/master/LICENSE
optree	https://optree.readthedocs.io/en/latest/	https://github.com/metaopt/optree/blob/main/LICENSE
oscrypto	https://github.com/wbond/oscrypto	https://github.com/wbond/oscrypto/blob/master/LICENSE
overrides	https://github.com/mkorpela/overrides	https://github.com/mkorpela/overrides/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
packaging	https://github.com/pypa/packaging	https://github.com/pypa/packaging/blob/main/LICENSE
packmol	https://leandro.iqm.unicamp.br/m3g/packmol/home.shtml	https://github.com/m3g/packmol/blob/master/LICENSE
packmol_memgen	http://cpclab.uni-duesseldorf.de/	https://gitlab.com/cbjh/memgen/py-memgen/-/blob/main/LICENSE
pandas	https://pandas.pydata.org	https://github.com/pandas-dev/pandas/blob/main/LICENSE
pandocfilters	http://github.com/jgm/pandocfilters	https://github.com/jgm/pandocfilters/blob/master/LICENSE
panedr	https://github.com/MDAnalysis/panedr	https://github.com/MDAnalysis/panedr/blob/master/LICENSE.txt
pango	https://pango.gnome.org	https://gitlab.gnome.org/GNOME/pango/-/blob/main/COPYING
ParmEd	https://parmed.github.io/ParmEd/html/index.html	https://github.com/ParmEd/ParmEd#license
parser	https://github.com/typescript-eslint/typescript-eslint	https://github.com/typescript-eslint/typescript-eslint/raw/master/LICENSE
parso	https://parso.readthedocs.io/en/latest/	https://github.com/davidhalter/parso/blob/master/LICENSE.txt
pathos	https://github.com/uqfoundation/pathos	https://github.com/uqfoundation/pathos/blob/master/LICENSE
patsy	https://patsy.readthedocs.io/en/latest/	https://github.com/pydata/patsy/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pbkdf2	https://golang.org/x/crypto/pbkdf2	https://cs.opensource.google/go/x/crypto/+/master:LICENSE
pbr	https://docs.openstack.org/pbr/latest/	https://opendev.org/openstack/pbr/src/branch/master/LICENSE
pcmsolver	https://pcmsolver.readthedocs.io/en/stable/	https://github.com/PCMSolver/pcmsolver/blob/master/LICENSE
pcre	https://www.pcre.org	https://www.pcre.org/licence.txt
pcre2	https://www.pcre.org	https://github.com/PCRE2Project/pcre2/blob/master/LICENCE
pdb4amber	https://github.com/Amber-MD/pdb4amber	https://github.com/Amber-MD/pdb4amber/blob/master/LICENSE
pdbfixer	https://github.com/openmm/pdbfixer	https://github.com/openmm/pdbfixer/blob/master/LICENSE
pexpect	https://pexpect.readthedocs.io/	https://github.com/pexpect/pexpect/blob/master/LICENSE
pgconn	https://github.com/jackc/pgconn	https://github.com/jackc/pgconn/blob/master/LICENSE
pgio	https://github.com/jackc/pgio	https://github.com/jackc/pgio/blob/master/LICENSE
pgpassfile	https://github.com/jackc/pgpassfile	https://github.com/jackc/pgpassfile/blob/master/LICENSE
pgproto3	https://github.com/jackc/pgproto3/v2	https://github.com/jackc/pgproto3/blob/master/LICENSE
pgtype	https://github.com/jackc/pgtype	https://github.com/jackc/pgtype/blob/master/LICENSE
pgx	https://github.com/jackc/pgx/v4	https://github.com/jackc/pgx/blob/v4.0.1/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
phonopy	https://github.com/phonopy/phono3py	https://github.com/phonopy/phono3py/blob/develop/LICENSE
pickleshare	https://github.com/pickleshare/pickleshare	https://github.com/pickleshare/pickleshare/blob/master/LICENSE
Pillow	https://python-pillow.org	https://github.com/python-pillow/Pillow/blob/main/LICENSE
Pint	https://pint.readthedocs.io/en/stable/	https://github.com/hgrecco/pint/blob/master/LICENSE
pip	https://pip.pypa.io/	https://github.com/pypa/pip/blob/main/LICENSE.txt
pip-licenses	https://github.com/raimon49/pip-licenses	https://github.com/raimon49/pip-licenses/blob/master/LICENSE
pixman	https://pixman.org	https://cgit.freedesktop.org/pixman/tree/COPYING
pkginfo	https://launchpad.net/pkginfo	https://bazaar.launchpad.net/~tseaver/pkginfo/trunk/view/head:/LICENSE.txt
pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name?tab=License-1-ov-file#readme
pkgutil_resolve_name	https://github.com/graingert/pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name?tab=License-1-ov-file#readme
platformdirs	https://github.com/platformdirs/platformdirs	https://github.com/platformdirs/platformdirs/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
plotly	https://plotly.com/python/	https://github.com/plotly/plotly.py/blob/master/LICENSE.txt
plotly-orca	https://github.com/plotly/orca	https://github.com/plotly/orca/blob/master/LICENSE
plotly.js	https://github.com/plotly/plotly.js	https://github.com/plotly/plotly.js/raw/master/LICENSE
pluggy	https://pluggy.readthedocs.io/en/stable/index.html	https://github.com/pytest-dev/pluggy/blob/main/LICENSE
pooch	https://github.com/fatiando/pooch	https://github.com/fatiando/pooch/blob/main/LICENSE.txt
portalocker	https://github.com/WoLpH/portalocker	https://github.com/wolph/portalocker/blob/develop/LICENSE
pox	https://github.com/uqfoundation/pox	https://github.com/uqfoundation/pox/blob/master/LICENSE
ppft	https://github.com/uqfoundation/ppft	https://github.com/uqfoundation/ppft/blob/master/LICENSE
pq	https://github.com/lib/pq	https://github.com/lib/pq/blob/master/LICENSE.md
ProDy	http://www.csb.pitt.edu/ProDy	https://github.com/prody/ProDy/blob/master/LICENSE.rst
prometheus-client	https://github.com/prometheus/client_python	https://github.com/prometheus/client_python/blob/master/LICENSE
prometheus_client	https://github.com/prometheus/client_python	https://github.com/prometheus/client_python/blob/master/LICENSE
prompt-toolkit	https://python-prompt-toolkit.readthedocs.io/en/stable/	https://github.com/prompt-toolkit/python-prompt-toolkit/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
prompt_toolkit	https://github.com/prompt-toolkit/python-prompt-toolki	https://github.com/prompt-toolkit/python-prompt-toolki tab=BSD-3-Clause-1-ov-f readme
propcache	https://github.com/aio-libs/propcache	https://github.com/aio-libs/propcache?tab=Apache-2.0-1-ov-file#readme
protobuf	https://google.golang.org/protobuf	https://github.com/protocolbuffers/protobuf-go/blob/master/LICENSE
psi4	https://psicode.org	https://github.com/psi4/psi4/blob/master/COPYING.LESSER
psutil	https://psutil.readthedocs.io/en/latest/	https://github.com/giampaolo/psutil/blob/master/LICENSE
psycopg2	https://psycopg.org/	https://github.com/psycopg/psycopg2/blob/master/LICENSE
PTable	https://github.com/kxxoling/PTable	https://github.com/kxxoling/PTable/blob/master/COPYING
pthread-stubs	https://xcb.freedesktop.org	https://github.com/freedesktop/xcb-pthread-stubs/blob/master/COPYING
ptyprocess	https://ptyprocess.readthedocs.io/en/latest/	https://github.com/pexpect/ptyprocess/blob/master/LICENSE
pure-eval	http://github.com/alexmojaki/pure_eval	http://github.com/alexmojaki/pure_eval/blob/master/LICENSE.txt
pure_eval	http://github.com/alexmojaki/pure_eval	https://github.com/alexmojaki/pure_eval?tab=MIT-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
py	https://py.readthedocs.io/en/latest/	https://github.com/pytest-dev/py/blob/master/LICENSE
py-cpuinfo	https://github.com/workhorsy/py-cpuinfo	https://github.com/workhorsy/py-cpuinfo/blob/master/LICENSE
pyasn1	https://github.com/etingof/pyasn1	https://github.com/etingof/pyasn1/blob/master/LICENSE.rst
pyasn1-modules	https://github.com/etingof/pyasn1-modules	https://github.com/etingof/pyasn1-modules/blob/master/LICENSE.txt
pyasn1_modules	https://pyasn1.readthedocs.io/en/latest/contents.html	https://github.com/pyasn1/pyasn1-modules?tab=BSD-2-Clause-1-ov-file#readme
pybind11	http://pybind11.readthedocs.io/en/stable/	https://github.com/pybind/pybind11/blob/master/LICENSE
pybind11-abi	https://github.com/pybind/pybind11	https://github.com/pybind/pybind11/blob/master/LICENSE
pybind11_global	http://pybind11.readthedocs.io/en/stable/	https://github.com/pybind/pybind11/blob/master/LICENSE
pycairo	https://pycairo.readthedocs.io/en/latest/index.html	https://github.com/pygobject/pycairo/blob/master/COPYING
pycosat	https://github.com/conda/pycosat	https://github.com/conda/pycosat/blob/main/LICENSE
pycparser	https://github.com/eliben/pycparser	https://github.com/eliben/pycparser/blob/master/LICENSE
pydantic	https://pydantic-docs.helpmanual.io	https://github.com/samuelcolvin/pydantic/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pydantic-core	https://github.com/pydantic/pydantic-core	https://github.com/pydantic/pydantic-core
pyedr	https://github.com/MDAnalysis/panedr	https://github.com/MDAnalysis/panedr/blob/master/LICENSE.txt
pyEMMA	http://github.com/markovmodel/PyEMMA	https://github.com/markovmodel/PyEMMA/blob/devel/LICENSE.txt
Pygments	https://pygments.org	https://github.com/pygments/pygments/blob/master/LICENSE
pygments	https://pygments.org/	https://github.com/pygments/pygments?tab=BSD-2-Clause-1-ov-file#readme
pygraphviz	https://pygraphviz.github.io	https://github.com/pygraphviz/pygraphviz/blob/main/LICENSE
pygtop	https://pygtop.readthedocs.io/en/latest/	https://github.com/samirelanduk/pygtop/blob/master/LICENSE
pyHanko	https://github.com/MatthiasValvekens/pyHanko	https://github.com/MatthiasValvekens/pyHanko/blob/master/LICENSE
pyhanko-certvalidator	https://github.com/MatthiasValvekens/certvalidator	https://github.com/MatthiasValvekens/certvalidator/blob/master/LICENSE
PyJWT	https://github.com/jpadilla/pyjwt	https://github.com/jpadilla/pyjwt/blob/master/LICENSE
pymbar	https://pymbar.org	https://github.com/choderalab/pymbar/blob/master/LICENSE
pynvml	https://github.com/gpuopenanalytics/pynvml	https://github.com/gpuopenanalytics/pynvml/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pyOpenSSL	https://pyopenssl.org/	https://github.com/pyca/pyopenssl/blob/main/LICENSE
pyparsing	https://pyparsing-docs.readthedocs.io/en/latest/	https://github.com/pyparsing/pyparsing/blob/master/LICENSE
PyPDF3	https://github.com/sfneal/PyPDF3	https://github.com/sfneal/PyPDF3/blob/master/LICENSE
pyrsistent	http://github.com/tobgu/pyrsistent/	https://github.com/tobgu/pyrsistent/blob/master/LICENSE.mit
pysam	https://github.com/pysam-developers/pysam	https://github.com/pysam-developers/pysam/blob/master/COPYING
PySocks	https://github.com/Anorov/PySocks	https://github.com/Anorov/PySocks/blob/master/LICENSE
pytables	https://www.pytables.org	https://github.com/PyTables/PyTables/blob/master/LICENSE.txt
python	https://www.python.org/	https://docs.python.org/3/license.html
python-bidi	https://github.com/MeirKriheli/python-bidi	https://github.com/MeirKriheli/python-bidi/blob/master/COPYING
python-constraint	https://github.com/python-constraint/python-constraint	https://github.com/python-constraint/python-constraint/blob/master/LICENSE
python-dateutil	https://dateutil.readthedocs.io	https://github.com/dateutil/dateutil/blob/master/LICENSE
python-json-logger	http://github.com/madzak/python-json-logger	https://github.com/madzak/python-json-logger/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
python-pptx	https://github.com/scanny/python-pptx	https://github.com/scanny/python-pptx/blob/master/LICENSE
python-ldap	https://www.python-ldap.org/	https://github.com/python-ldap/python-ldap/blob/python-ldap-3.4.0/LICENCE
python3-saml	https://github.com/onelogin/python3-saml	https://github.com/onelogin/python3-saml/blob/master/LICENSE
python_abi	https://github.com/conda-forge/python_abi-feedstock	https://github.com/conda-forge/python_abi-feedstock/blob/main/LICENSE.txt
pytorch-lightning	https://github.com/Lightning-AI/pytorch-lightning	https://github.com/Lightning-AI/pytorch-lightning/blob/master/LICENSE
pytz	https://pythonhosted.org/pytz	https://git.launchpad.net/pytz/tree/LICENSE.txt
pytz-deprecation-shim	https://github.com/pganssle/pytz-deprecation-shim	https://github.com/pganssle/pytz-deprecation-shim/blob/master/LICENSE
PyWavelets	https://github.com/PyWavelets/pywt	https://github.com/PyWavelets/pywt/blob/master/LICENSE
PyYAML	https://pyyaml.org/	https://github.com/yaml/pyyaml/blob/master/LICENSE
pyyml	No longer available	No longer available
pyzmq	https://pyzmq.readthedocs.io/en/latest/	https://github.com/zeromq/pyzmq/blob/main/COPYING.BSD
qcelemental	https://github.com/MolSSI/QCElemental	https://github.com/MolSSI/QCElemental/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
qcengine	https://github.com/MolSSI/QCEngine	https://github.com/MolSSI/QCEngine/blob/master/LICENSE
qrcode	https://github.com/lincolnloop/python-qrcode	https://github.com/lincolnloop/python-qrcode/blob/main/LICENSE
ramda	https://github.com/ramda/ramda	https://github.com/ramda/ramda/raw/master/LICENSE.txt
rapidjson	https://rapidjson.org/	https://github.com/Tencent/rapidjson/blob/master/license.txt
rdkit	https://www.rdkit.org	https://github.com/rdkit/rdkit/blob/master/license.txt
re2	https://github.com/google/re2	https://github.com/google/re2/blob/main/LICENSE
readme-renderer	https://github.com/pypa/readme_renderer	https://github.com/pypa/readme_renderer/blob/main/LICENSE
redis-py	https://github.com/andymccurdy/redis-py	https://github.com/redis/redis-py/blob/master/LICENSE
referencing	https://github.com/python-jsonschema/referencing	https://github.com/python-jsonschema/referencing
regex	https://github.com/mrabarnett/mrab-regex	https://github.com/mrabarnett/mrab-regex/blob/hg/LICENSE.txt
reportlab	https://www.reportlab.com	https://www.reportlab.com/dev/docs/faq/#1.3
reproc	https://github.com/DaanDeMeyer/reproc	https://github.com/DaanDeMeyer/reproc/blob/main/LICENSE
reproc-cpp	https://github.com/DaanDeMeyer/reproc	https://github.com/DaanDeMeyer/reproc/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
requests	https://requests.readthedocs.io	https://github.com/psf/requests/blob/main/LICENSE
requests-oauthlib	https://github.com/requests/requests-oauthlib	https://github.com/requests/requests-oauthlib/blob/master/LICENSE
requests-toolbelt	https://toolbelt.readthedocs.org	https://github.com/requests/toolbelt/blob/master/LICENSE
resumable	https://github.com/stevvooe/resumable	https://github.com/stevvooe/resumable/blob/master/LICENSE
retrying	https://github.com/rholder/retrying	https://github.com/rholder/retrying/blob/master/LICENSE
rfc3339-validator	https://github.com/naimetti/rfc3339-validator	https://github.com/naimetti/rfc3339-validator/blob/master/LICENSE
rfc3339_validator	https://github.com/naimetti/rfc3339-validator	https://github.com/naimetti/rfc3339-validator?tab=MIT-1-ov-file#readme
rfc3986	https://rfc3986.readthedocs.io/en/latest/	https://github.com/python-hyper/rfc3986/blob/main/LICENSE
rfc3986-validator	https://github.com/naimetti/rfc3986-validator	https://github.com/naimetti/rfc3986-validator/blob/master/LICENSE
rfc3987-syntax	https://github.com/willynilly/rfc3987-syntax	https://github.com/willynilly/rfc3987-syntax?tab=MIT-1-ov-file#readme
rich	https://github.com/Textualize/rich	https://github.com/Textualize/rich/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
rich-argparse	https://github.com/hamdanal/rich-argparse	https://github.com/hamdanal/rich-argparse/blob/main/LICENSE
rlPyCairo	https://pypi.org/project/rlPyCairo/	https://hg.reportlab.com/hg-public/rlPyCairo/file/tip/rlPyCairo/LICENSE.txt
RocksDict	https://github.com/rocksdict/RocksDict	https://github.com/rocksdict/RocksDict?tab=MIT-1-ov-file#readme
roman-numerals	https://github.com/AA-Turner/roman-numerals	https://github.com/AA-Turner/roman-numerals?tab=License-1-ov-file#readme
roman-numerals-py	https://github.com/AA-Turner/roman-numerals	https://github.com/AA-Turner/roman-numerals?tab=License-1-ov-file#readme
rotary-embedding-torch	https://github.com/lucidrains/rotary-embedding-tor	https://github.com/lucidrains/rotary-embedding-tor blob/main/LICENSE
rpds-py	https://github.com/crate-py/rpds	https://github.com/crate-py/rpds
rpmpack	https://github.com/google/rpmpack	https://github.com/google/rpmpack/blob/master/LICENSE
rsa	https://stuvel.eu/rsa	https://github.com/sybrenstuvel/python-rsa/blob/main/LICENSE
ruamel-yaml	https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/	https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/LICENSE
ruamel.yaml.clib	https://sourceforge.net/p/ruamel-yaml-clib/code/ci/default/tree/	https://sourceforge.net/p/ruamel-yaml-clib/code/ci/default/tree/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
s3transfer	https://github.com/boto/s3transfer	https://github.com/boto/s3transfer/blob/develop/LICENSE.txt
sacrebleu	https://github.com/mjpost/sacrebleu	https://github.com/mjpost/sacrebleu/blob/master/LICENSE.txt
sapiens	https://github.com/Merck/Sapiens	https://github.com/Merck/Sapiens/blob/main/LICENSE
sasl	https://mellium.im/sasl	https://github.com/mellium/sasl/blob/main/LICENSE
scikit-gstat	https://mmaelicke.github.io/scikit-gstat/	https://github.com/mmaelicke/scikit-gstat/blob/main/LICENSE
scikit-image	https://scikit-image.org	https://scikit-image.org/docs/dev/license.html
scikit-learn	https://scikit-learn.org/stable/	https://github.com/scikit-learn/scikit-learn/blob/main/COPYING
scikit-learn-extra	https://github.com/scikit-learn-contrib/scikit-learn-extra	https://github.com/scikit-learn-contrib/scikit-learn-extra/blob/main/LICENSE
scikit-network	https://github.com/sknetwork-team/scikit-network	https://github.com/sknetwork-team/scikit-network?tab=License-1-ov-file#readme
scipy	https://scipy.org	https://github.com/scipy/scipy/blob/main/LICENSE.txt
seaborn	https://seaborn.pydata.org	https://github.com/mwaskom/seaborn/blob/master/LICENSE
seaborn-base	https://seaborn.pydata.org	https://github.com/mwaskom/seaborn/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
semver	https://github.com/Masterminds/semver/v3	https://github.com/Masterminds/semver/blob/v3.0.3/LICENSE.txt
Send2Trash	https://github.com/arsenetar/send2trash	https://github.com/arsenetar/send2trash/blob/master/LICENSE
sentry-sdk	https://github.com/getsentry/sentry-python	https://github.com/getsentry/sentry-python/blob/master/LICENSE
setproctitle	https://github.com/dvarrazzo/py-setproctitle	https://github.com/dvarrazzo/py-setproctitle/blob/master/LICENSE
setuptools	https://github.com/pypa/setuptools	https://github.com/pypa/setuptools/blob/main/LICENSE
setuptools-scm	https://github.com/pypa/setuptools_scm/	https://github.com/pypa/setuptools_scm/blob/main/LICENSE
sh	https://github.com/amoffat/sh	https://github.com/amoffat/sh/blob/develop/LICENSE.txt
shellingham	https://github.com/sarugaku/shellingham	https://github.com/sarugaku/shellingham/blob/master/LICENSE
simdjson	https://github.com/simdjson/simdjson/tree/master	https://github.com/simdjson/simdjson/blob/master/LICENSE
simint	https://www.bennyp.org/research/simint/	https://github.com/simint-chem/simint-generator/blob/master/LICENSE
simplejson	https://github.com/simplejson/simplejson	https://github.com/simplejson/simplejson/blob/master/LICENSE.txt
six	https://github.com/benjaminp/six	https://github.com/benjaminp/six/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
smirnoff99frosst	https://github.com/openforcefield/smirnoff99frosst	https://github.com/openforcefield/smirnoff99Frosst/blob/master/LICENSE
smmap	https://github.com/gitpython-developers/smmap	https://github.com/gitpython-developers/smmap/blob/master/LICENSE
snappy	https://github.com/google/snappy	https://github.com/google/snappy/blob/main/COPYING
sniffio	https://github.com/python-trio/sniffio	https://github.com/python-trio/sniffio/blob/master/LICENSE.MIT
snowballstemmer	https://github.com/snowballstem/snowball	https://github.com/snowballstem/snowball/blob/master/COPYING
soupsieve	https://github.com/facelessuser/soupsieve	https://github.com/facelessuser/soupsieve/blob/main/LICENSE.md
spdlog	https://github.com/gabime/spdlog	https://github.com/gabime/spdlog?tab=License-1-ov-file#readme
spglib	https://spglib.readthedocs.io/	https://github.com/spglib/spglib/blob/develop/COPYING
sphinx	https://github.com/sphinx-doc/sphinx	https://github.com/sphinx-doc/sphinx/blob/5.x/LICENSE
sphinxcontrib-applehelp	https://github.com/sphinx-doc/sphinxcontrib-applehe	https://github.com/sphinx-doc/sphinxcontrib-applehe blob/master/LICENSE
sphinxcontrib-devhelp	https://github.com/sphinx-doc/sphinxcontrib-devhelp	https://github.com/sphinx-doc/sphinxcontrib-devhelp blob/master/LICENSE
sphinxcontrib-htmlhelp	https://github.com/sphinx-doc/sphinxcontrib-htmlhel	https://github.com/sphinx-doc/sphinxcontrib-htmlhel blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
sphinxcontrib-jsmath	https://github.com/sphinx-doc/sphinxcontrib-jsmath	https://github.com/sphinx-doc/sphinxcontrib-jsmath/blob/master/LICENSE
sphinxcontrib-qthelp	https://github.com/sphinx-doc/sphinxcontrib-qthelp	https://github.com/sphinx-doc/sphinxcontrib-qthelp/blob/master/LICENSE
sphinxcontrib-serializinghtml	https://github.com/sphinx-doc/sphinxcontrib-serializ	https://github.com/sphinx-doc/sphinxcontrib-serializ blob/master/LICENSE
SQLAlchemy	https://www.sqlalchemy.org	https://docs.sqlalchemy.org/en/14/copyright.html
SQLAlchemy	https://www.sqlalchemy.org/	https://github.com/sqlalchemy/sqlalchemy?tab=MIT-1-ov-file#readme
sqlite	https://sqlite.org/index.html	https://sqlite.org/copyright.html
sqlparse	https://github.com/andialbrecht/sqlparse	https://github.com/andialbrecht/sqlparse/blob/master/LICENSE
stack-data	http://github.com/alexmojaki/stack_data	https://github.com/alexmojaki/stack_data/blob/master/LICENSE.txt
stack_data	http://github.com/alexmojaki/stack_data	https://github.com/alexmojaki/stack_data?tab=MIT-1-ov-file#readme
starfile	https://github.com/alisterburt/starfile	https://github.com/teamtomo/starfile/blob/main/LICENSE
statsmodels	https://github.com/statsmodels/statsmodels	https://github.com/statsmodels/statsmodels/blob/main/LICENSE.txt
structlog	https://www.structlog.org/	https://github.com/hynek/structlog/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
streamlit	https://github.com/streamlit/streamlit	https://github.com/streamlit/streamlit/blob/develop/LICENSE
svglib	https://github.com/deeplook/svglib	https://github.com/deeplook/svglib/blob/master/LICENSE.txt
sympy	https://sympy.org	https://github.com/sympy/sympy/blob/master/LICENSE
tables	https://www.pytables.org/	https://github.com/PyTables/PyTables/blob/master/LICENSE.txt
tabulate	https://github.com/astanin/python-tabulate	https://github.com/astanin/python-tabulate/blob/master/LICENSE
tbb	https://github.com/oneapi-src/oneTBB	https://github.com/oneapi-src/oneTBB/blob/master/LICENSE.txt
tenacity	https://github.com/jd/tenacity	https://github.com/jd/tenacity/blob/master/LICENSE
tensorboard	https://github.com/tensorflow/tensorboard	https://github.com/tensorflow/tensorboard/blob/master/LICENSE
tensorboard-data-server	https://github.com/tensorflow/tensorboard	https://github.com/tensorflow/tensorboard/blob/master/LICENSE
tensorboard-plugin-wit	https://github.com/pair-code/what-if-tool	https://github.com/PAIR-code/what-if-tool/blob/master/LICENSE
tensorboardX	https://github.com/lanpa/tensorboardX	https://github.com/lanpa/tensorboardX/blob/master/LICENSE
tensorflow	https://github.com/tensorflow/tensorflow	https://github.com/tensorflow/tensorflow/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
tensorflow-estimator	https://github.com/tensorflow/estimator	https://github.com/tensorflow/estimator/blob/master/LICENSE
tensorflow-io-gcs-filesystem	https://github.com/tensorflow/io	https://github.com/tensorflow/io/blob/master/LICENSE
tensorflow-probability	https://github.com/tensorflow/probability	https://github.com/tensorflow/probability/blob/main/LICENSE
termcolor	https://github.com/hugovk/termcolor	https://github.com/hugovk/termcolor/blob/main/COPYING.txt
terminado	https://github.com/jupyter/terminado	https://github.com/jupyter/terminado/blob/main/LICENSE
testpath	https://github.com/jupyter/testpath	https://github.com/jupyter/testpath/blob/master/LICENSE
textangular	https://github.com/fraywing/textAngular	https://github.com/fraywing/textAngular
texttable	https://github.com/foutaise/texttable/	https://github.com/foutaise/texttable/?tab=MIT-1-ov-file#readme
tf_keras	https://github.com/keras-team/keras-io	https://github.com/keras-team/keras-io/blob/master/LICENSE
tf_keras	https://github.com/keras-team/keras-io	https://github.com/keras-team/keras-io/blob/master/LICENSE
threadpoolctl	https://github.com/joblib/threadpoolctl	https://github.com/joblib/threadpoolctl/blob/master/LICENSE
three	https://github.com/mrdoob/three.js	https://github.com/mrdoob/three.js/raw/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
tiffile	https://github.com/cgohlke/tiffile/	https://github.com/cgohlke/tiffile/blob/master/LICENSE
tinycss2	https://github.com/Kozea/tinycss2/	https://github.com/Kozea/tinycss2/blob/master/LICENSE
tinyxml2	https://github.com/leethomason/tinyxml2	https://github.com/leethomason/tinyxml2/blob/master/LICENSE.txt
tk	https://www.tcl.tk/	https://www.tcl.tk/software/tcltk/license.html
toml	https://github.com/toml-lang/toml	https://github.com/toml-lang/toml/blob/main/LICENSE
tomli	https://github.com/hukkin/tomli	https://github.com/hukkin/tomli/blob/master/LICENSE
toolz	https://github.com/pytoolz/toolz	https://github.com/pytoolz/toolz/blob/master/LICENSE.txt
torch	https://pytorch.org/	https://pypi.org/project/torch/#license
torch-geometric	https://pyg.org/	https://github.com/pyg-team/pytorch_geometric?tab=MIT-1-ov-file#readme
torcheval	https://github.com/pytorch/torcheval	https://github.com/pytorch/torcheval?tab=License-1-ov-file#readme
torchmetrics	https://github.com/Lightning-AI/torchmetrics	https://github.com/Lightning-AI/torchmetrics?tab=Apache-2.0-1-ov-file#readme
tornado	https://www.tornadoweb.org	https://github.com/tornadoweb/tornado/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL		
tqdm	https://github.com/tqdm/tqdm	https://github.com/tqdm/tqdm/blob/master/LICENCE		
tracy	https://github.com/gear-genomics/tracy	https://github.com/gear-genomics/tracy/blob/main/LICENSE		
traitlets	https://github.com/ipython/traitlets	https://github.com/ipython/traitlets/blob/main/COPYING.md		
triton	https://github.com/openai/triton/	https://github.com/openai/triton/blob/main/LICENSE		
truststore	https://github.com/sethmlarson/trus	https://github.com/sethmlarson/truststore/blob/main/LICENSE		
truststore	https://github.com/sethmlarson/truststore	https://github.com/sethmlarson/truststore/blob/main/LICENSEtruststore	https://github.com/sethmlarson/truststore	https://github.com/sethmlarson/truststore/blob/main/LICENSE
ts-jest	https://github.com/kulshekhar/ts-jest	https://github.com/kulshekhar/ts-jest/raw/master/LICENSE.md		
ts-loader	https://github.com/TypeStrong/ts-loader	https://github.com/TypeStrong/ts-loader/raw/master/LICENSE		
twine	https://github.com/pypa/twine	https://github.com/pypa/twine/blob/main/LICENSE		
twinj uuid	https://github.com/twinj/uuid	https://github.com/twinj/uuid/blob/master/LICENSE		
types	https://github.com/babel/babel	https://github.com/babel/babel/raw/master/LICENSE		
types-python-dateutil	https://github.com/python/typeshed	https://github.com/python/typeshed?tab=License-1-ov-file#readme		
types-requests	https://pypi.org/project/types-requests/	https://github.com/python/typeshed/blob/main/LICENSE		

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
typescript	https://github.com/Microsoft/TypeScript	https://github.com/Microsoft/TypeScript/raw/master/LICENSE.txt
typing-inspection	https://github.com/pydantic/typing-inspection	https://github.com/pydantic/typing-inspection?tab=MIT-1-ov-file#readme
typing-utils	https://github.com/bojiang/typing_utils	https://github.com/bojiang/typing_utils?tab=Apache-2.0-1-ov-file#readme
typing_extensions	https://github.com/python/typing	https://github.com/python/typing/blob/master/LICENSE
typing_utils	https://github.com/bojiang/typing_utils	https://github.com/bojiang/typing_utils?tab=Apache-2.0-1-ov-file#readme
tzdata	https://github.com/python/tzdata	https://github.com/python/tzdata/blob/master/LICENSE
tzlocal	https://github.com/regebro/tzlocal	https://github.com/regebro/tzlocal/blob/master/LICENSE.txt
umf	https://github.com/oneapi-src/unified-memory-fram	https://github.com/oneapi-src/unified-memory-fram tab=License-1-ov-file#readme
umi-tools	https://github.com/CGATOxford/UMI-tools	https://github.com/CGATOxford/UMI-tools/blob/master/LICENSE
unicodedata2	https://github.com/fonttools/unicodedata2	https://github.com/fonttools/unicodedata2/blob/master/LICENSE
uri-template	https://gitlab.linss.com/open-source/python/uri-template	https://gitlab.linss.com/open-source/python/uri-template/-/blob/main/LICENSE
uritools	https://github.com/tkem/uritools/	https://github.com/tkem/uritools/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL		
urllib3	https://urllib3.readthedocs.io/	https://github.com/urllib3/urllib3/blob/main/LICENSE.txt		
vine	https://github.com/celery/vine	https://github.com/celery/vine/blob/master/LICENSE		
vue	https://github.com/vuejs/vue	https://github.com/vuejs/vue/raw/master/LICENSE		
wadler_lindig	https://github.com/patrick-kidger/wadler_lindig	https://github.com/patrick-kidger/wadler_lindig?tab=Apache-2.0-1-ov-file#readme		
wadler_lindig	https://github.com/patrick-kidger/wadler_lindig	https://github.com/patrick-kidger/wadler_lindig?tab=Apache-2.0-1-ov-file#readmewcwidth	https://github.com/jquast/wcwidth	https://github.com/jquast/wcwidth/blob/master/LICENSE
wandb	https://github.com/wandb/wandb	https://github.com/wandb/wandb/blob/main/LICENSE		
webcolors	https://github.com/ubernostrum/webcolors	https://github.com/ubernostrum/webcolors?tab=BSD-3-Clause-1-ov-file#readme		
webencodings	https://github.com/gsnedders/python-webencodings	https://github.com/gsnedders/python-webencodings/blob/master/LICENSE		
websocket-client	https://github.com/websocket-client/websocket-client.git	https://github.com/websocket-client/websocket-client/blob/master/LICENSE		
Werkzeug	https://palletsprojects.com/p/werkzeug/	https://werkzeug.palletsprojects.com/en/2.1.x/license/		
westpa	http://github.com/westpa/westpa	https://github.com/westpa/westpa/blob/westpa2/LICENSE		
wheel	https://github.com/pypa/wheel	https://github.com/pypa/wheel/blob/main/LICENSE.txt		

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
widgetsnbextension	https://github.com/jupyter-widgets/ipywidgets#readme	https://github.com/jupyter-widgets/ipywidgets/blob/main/LICENSE
wrapt	https://github.com/GrahamDumpleton/wrapt	https://github.com/GrahamDumpleton/wrapt/blob/develop/LICENSE
wsutil	https://github.com/yhat/wsutil	https://github.com/yhat/wsutil/blob/master/LICENSE
x/lint	https://golang.org/x/lint	https://github.com/golang/lint/blob/master/LICENSE
x/mod	https://golang.org/x/mod/semver	https://cs.opensource.google/go/x/mod/+/master:LICENSE
x/net	https://golang.org/x/net	https://cs.opensource.google/go/x/net/+/master:LICENSE
x/oauth2	https://golang.org/x/oauth2	https://cs.opensource.google/go/x/oauth2/+/master:LICENSE
x/sys	https://golang.org/x/sys	https://cs.opensource.google/go/x/sys/+/master:LICENSE
x/text	https://golang.org/x/text	https://cs.opensource.google/go/x/text/+/master:LICENSE
x/tools	https://golang.org/x/tools	https://cs.opensource.google/go/x/tools/+/master:LICENSE
x/xerrors	https://golang.org/x/xerrors	https://cs.opensource.google/go/x/xerrors/+/master:LICENSE
xgboost	https://github.com/dmlc/xgboost	https://github.com/dmlc/xgboost/blob/master/LICENSE
xhtml2pdf	http://github.com/xhtml2pdf/xhtml2pdf	https://github.com/xhtml2pdf/xhtml2pdf/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
xlrd	https://github.com/python-excel/xlrd	https://github.com/python-excel/xlrd/blob/master/LICENSE
xlsxwriter	https://github.com/jmcnamara/XlsxWriter	https://github.com/jmcnamara/XlsxWriter/blob/main/LICENSE.txt
xmlsec	https://github.com/mehcode/python-xmlsec	https://github.com/mehcode/python-xmlsec/blob/master/LICENSE
xmltodict	https://github.com/martinblech/xmltodict	https://github.com/martinblech/xmltodict/blob/master/LICENSE
xorg-kbproto	https://gitlab.freedesktop.org/xorg/proto/kbproto	https://gitlab.freedesktop.org/xorg/proto/kbproto/-/blob/master/COPYING
xorg-libice	https://gitlab.freedesktop.org/xorg/lib/libice	https://gitlab.freedesktop.org/xorg/lib/libice/-/blob/master/COPYING
xorg-libsm	https://gitlab.freedesktop.org/xorg/lib/libsm	https://gitlab.freedesktop.org/xorg/lib/libsm/-/blob/master/COPYING
xorg-libx11	https://gitlab.freedesktop.org/xorg/lib/libx11	https://gitlab.freedesktop.org/xorg/lib/libx11/-/blob/master/COPYING
xorg-libxau	https://gitlab.freedesktop.org/xorg/lib/libxau	https://gitlab.freedesktop.org/xorg/lib/libxau/-/blob/master/COPYING
xorg-libxdmcp	https://gitlab.freedesktop.org/xorg/lib/libxdmcp	https://gitlab.freedesktop.org/xorg/lib/libxdmcp/-/blob/master/COPYING

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
xorg-libxext	https://gitlab.freedesktop.org/xorg/lib/libxext	https://gitlab.freedesktop.org/xorg/lib/libxext/-/blob/master/COPYING
xorg-libxrender	https://gitlab.freedesktop.org/xorg/lib/libxrender	https://gitlab.freedesktop.org/xorg/lib/libxrender/-/blob/master/COPYING
xorg-libxt	https://gitlab.freedesktop.org/xorg/lib/libxt	https://gitlab.freedesktop.org/xorg/lib/libxt/-/blob/master/COPYING
xorg-renderproto	https://gitlab.freedesktop.org/xorg/proto/renderproto	https://gitlab.freedesktop.org/xorg/proto/renderproto/-/blob/master/COPYING
xorg-xextproto	https://gitlab.freedesktop.org/xorg/proto/xextproto	https://gitlab.freedesktop.org/xorg/proto/xextproto/-/blob/master/COPYING
xorg-xproto	https://gitlab.freedesktop.org/xorg/proto/xproto	https://gitlab.freedesktop.org/xorg/proto/xproto/-/blob/master/COPYING
xxhash	https://github.com/cespare/xxhash/v2	https://github.com/cespare/xxhash/blob/main/LICENSE.txt
xz	https://github.com/ulikunitz/xz	https://github.com/ulikunitz/xz/blob/master/LICENSE
yaml	https://pyyaml.org/	https://github.com/yaml/pyyaml/blob/master/LICENSE
yaml-cpp	https://github.com/jbeder/yaml-cpp	https://github.com/jbeder/yaml-cpp/blob/master/LICENSE
yaml.v2	https://gopkg.in/yaml.v2	https://github.com/go-yaml/yaml/blob/v2.4.0/LICENSE
yaml.v3	https://gopkg.in/yaml.v3	https://github.com/go-yaml/yaml/blob/v3/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
yarl	https://github.com/aio-libs/yarl/	https://github.com/aio-libs/yarl/blob/master/LICENSE
yaspin	https://github.com/pavdmyt/yaspin	https://github.com/pavdmyt/yaspin/blob/master/LICENSE
yfiles	https://www.yworks.com/products/yfiles	commercial license
yml	https://pypi.org/project/yml/	N/A
zap	https://go.uber.org/zap	https://github.com/uber-go/zap/blob/master/LICENSE.txt
zipp	https://github.com/jaraco/zipp	https://github.com/jaraco/zipp/blob/main/LICENSE
zlib	https://zlib.net/	https://zlib.net/zlib_license.html
zstandard	https://github.com/indygreg/python-zstandard	https://github.com/indygreg/python-zstandard/blob/main/LICENSE
zstd	https://facebook.github.io/zstd/	https://github.com/facebook/zstd/blob/dev/LICENSE
click-option-group	https://github.com/click-contrib/click-option-group	https://github.com/click-contrib/click-option-group/blob/master/LICENSE

10.1 GCC

On the Linux platform, The OpenEye toolkits are linked with the GCC libraries using an eligible compilation process under the terms of the GCC RUNTIME LIBRARY EXCEPTION (see below).

The OpenEye toolkits are NOT open source software and are NOT governed by GPL or other open source licenses. Your right to use the toolkits are governed by the OpenEye license which is binding to You.

For certain Linux distributions, GCC library components are included as object files. Your rights to use GCC library components (libstdc++, libgcc, and libgomp) included with these distributions are governed by the GPLv3 license as follows:

The components libstdc++, libgcc, and libgomp are part of GCC.

- GCC is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
- GCC is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the

implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with GCC (see below). If not, see Licenses GNU Project .

Per GPLv3, section 6d, the source for GCC is available for download from the following location: GNU. The GCC library components included here are from version gcc-4.9.3 and are built using the standard instructions at Installing GCC.

10.1.1 GCC RUNTIME LIBRARY EXCEPTION

```
Version 3.1, 31 March 2009

Copyright © 2009 Free Software Foundation, Inc. http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies of this license document,␣
↪but changing it is not allowed.

This GCC Runtime Library Exception ("Exception") is an additional permission under␣
↪section 7 of the GNU General Public License,
version 3 ("GPLv3"). It applies to a given file (the "Runtime Library") that bears a␣
↪notice placed by the copyright holder
of the file stating that the file is governed by GPLv3 along with this Exception.

When you use GCC to compile a program, GCC may combine portions of certain GCC header␣
↪files and runtime libraries with
the compiled program. The purpose of this Exception is to allow compilation of non-GPL␣
↪(including proprietary) programs
to use, in this way, the header files and runtime libraries covered by this Exception.

0. Definitions.

A file is an "Independent Module" if it either requires the Runtime Library for␣
↪execution after a Compilation Process,
or makes use of an interface provided by the Runtime Library, but is not otherwise based␣
↪on the Runtime Library.

"GCC" means a version of the GNU Compiler Collection, with or without modifications,␣
↪governed by version 3 (or a specified
later version) of the GNU General Public License (GPL) with the option of using any␣
↪subsequent versions published by the FSF.

"GPL-compatible Software" is software whose conditions of propagation, modification and␣
↪use would permit combination with
GCC in accord with the license of GCC.

"Target Code" refers to output from any compiler for a real or virtual target processor␣
↪architecture, in executable
form or suitable for input to an assembler, loader, linker and/or execution phase.␣
↪Notwithstanding that, Target Code
does not include data in any format that is used as a compiler intermediate␣
↪representation, or used for producing a
compiler intermediate representation.
```

(continues on next page)

(continued from previous page)

```
The "Compilation Process" transforms code entirely represented in non-intermediate␣
↪languages designed for human-written
code, and/or in Java Virtual Machine byte code, into Target Code. Thus, for example, use␣
↪of source code generators and
preprocessors need not be considered part of the Compilation Process, since the␣
↪Compilation Process can be understood
as starting with the output of the generators or preprocessors.

A Compilation Process is "Eligible" if it is done using GCC, alone or with other GPL-
↪compatible software, or if it is
done without using any work based on GCC. For example, using non-GPL-compatible Software␣
↪to optimize any GCC intermediate
representations would not qualify as an Eligible Compilation Process.

1. Grant of Additional Permission.

You have permission to propagate a work of Target Code formed by combining the Runtime␣
↪Library with Independent Modules,
even if such propagation would otherwise violate the terms of GPLv3, provided that all␣
↪Target Code was generated by
Eligible Compilation Processes. You may then convey such a combination under terms of␣
↪your choice, consistent with the
licensing of the Independent Modules.

2. No Weakening of GCC Copyleft.

The availability of this Exception does not imply any general presumption that third-
↪party software is unaffected by
the copyleft requirements of the license of GCC.
```

See also

- http://www.gnu.org/licenses/gcc-exception.html

10.1.2 GNU GENERAL PUBLIC LICENSE

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

 The GNU General Public License is a free, copyleft license for
 software and other kinds of works.

   The licenses for most software and other practical works are designed
```

(continues on next page)

(continued from previous page)

```
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

  To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights.  Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received.  You must make sure that they, too, receive
or can get the source code.  And you must show them these terms so they
know their rights.

  Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

  For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software.  For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

  Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so.  This is fundamentally incompatible with the aim of
protecting users' freedom to change the software.  The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable.  Therefore, we
have designed this version of the GPL to prohibit the practice for those
products.  If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

  Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary.  To prevent this, the GPL assures that
```

(continues on next page)

(continued from previous page)

```
patents cannot be used to render the program non-free.

  The precise terms and conditions for copying, distribution and
modification follow.

                    TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.

  To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

  A "covered work" means either the unmodified Program or a work based
on the Program.

  To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

  1. Source Code.

  The "source code" for a work means the preferred form of the work
for making modifications to it.  "Object code" means any non-source
form of a work.
```

(continues on next page)

(continued from previous page)

```
  A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
```

(continues on next page)

(continued from previous page)

```
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.


  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

   a) The work must carry prominent notices stating that you modified
   it, and giving a relevant date.

   b) The work must carry prominent notices stating that it is
   released under this License and any conditions added under section
   7.  This requirement modifies the requirement in section 4 to
   "keep intact all notices".
```

(continues on next page)

(continued from previous page)

```
    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy.  This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged.  This License gives no
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

  a) Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by the
  Corresponding Source fixed on a durable physical medium
  customarily used for software interchange.

  b) Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by a
  written offer, valid for at least three years and valid for as
  long as you offer spare parts or customer support for that product
  model, to give anyone who possesses the object code either (1) a
  copy of the Corresponding Source for all the software in the
  product that is covered by this License, on a durable physical
  medium customarily used for software interchange, for a price no
  more than your reasonable cost of physically performing this
  conveying of source, or (2) access to copy the
  Corresponding Source from a network server at no charge.

  c) Convey individual copies of the object code with a copy of the
  written offer to provide the Corresponding Source.  This
  alternative is allowed only occasionally and noncommercially, and
  only if you received the object code with such an offer, in accord
```

(continues on next page)

(continued from previous page)

```
   with subsection 6b.

   d) Convey the object code by offering access from a designated
   place (gratis or for a charge), and offer equivalent access to the
   Corresponding Source in the same way through the same place at no
   further charge.  You need not require recipients to copy the
   Corresponding Source along with the object code.  If the place to
   copy the object code is a network server, the Corresponding Source
   may be on a different server (operated by you or a third party)
   that supports equivalent copying facilities, provided you maintain
   clear directions next to the object code saying where to find the
   Corresponding Source.  Regardless of what server hosts the
   Corresponding Source, you remain obligated to ensure that it is
   available for as long as needed to satisfy these requirements.

   e) Convey the object code using peer-to-peer transmission, provided
   you inform other peers where the object code and Corresponding
   Source of the work are being offered to the general public at no
   charge under subsection 6d.

  A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

  A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling.  In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage.  For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product.  A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

  "Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source.  The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.

  If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information.  But this requirement does not apply
```

(continues on next page)

(continued from previous page)

```
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

  The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

  Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

    a) Disclaiming warranty or limiting liability differently from the
    terms of sections 15 and 16 of this License; or

    b) Requiring preservation of specified reasonable legal notices or
    author attributions in that material or in the Appropriate Legal
    Notices displayed by works containing it; or

    c) Prohibiting misrepresentation of the origin of that material, or
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
```

(continues on next page)

(continued from previous page)

```
   authors of the material; or

   e) Declining to grant rights under trademark law for use of some
   trade names, trademarks, or service marks; or

   f) Requiring indemnification of licensors and authors of that
   material by anyone who conveys the material (or modified versions of
   it) with contractual assumptions of liability to the recipient, for
   any liability that these contractual assumptions directly impose on
   those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.
```

(continues on next page)

(continued from previous page)

```
  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License.  You are not responsible
for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations.  If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License.  For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based.  The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
```

(continues on next page)

```
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version.  For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement).  To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

  If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

  A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
```

(continues on next page)

(continued from previous page)

```
or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all.  For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

  13. Use with the GNU Affero General Public License.

  Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work.  The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time.  Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

  Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

  Later license versions may give you additional or different
permissions.  However, no additional obligations are imposed on any
```

(continues on next page)

(continued from previous page)

```
author or copyright holder as a result of your choosing to follow a
later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
```

(continues on next page)

(continued from previous page)

```
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs.  If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.  But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.
```

See also

- http://www.gnu.org/licenses/gpl.txt

INDEX

A

AddCurveSegment
    OEDepict::OE2DPath, 204
AddLineSegment
    OEDepict::OE2DPath, 205
AddStartPoint
    OEDepict::OE2DPath, 205
AddSVGClass
    OEDepict::OESVGGroup, 414

C

Clear
    OEDepict::OEAlignmentResult, 217
    OEDepict::OEImage, 286
    OEDepict::OEImageBase, 295
    OEDepict::OEImageFrame, 316
Constructors
    OEDepict::OE2DAtomDisplay, 123
    OEDepict::OE2DBondDisplay, 133
    OEDepict::OE2DMolDisplay, 140
    OEDepict::OE2DMolDisplayOptions, 147
    OEDepict::OE2DPath, 203
    OEDepict::OE2DPathPoint, 207
    OEDepict::OE2DPoint, 208
    OEDepict::OEAlignmentOptions, 210
    OEDepict::OEAlignmentResult, 216
    OEDepict::OEAtomSVGAtomIdxMarkup, 219
    OEDepict::OEAtomSVGNoMarkup, 222
    OEDepict::OEAtomSVGResidueMarkup, 223
    OEDepict::OEBondSVGBondIdxMarkup, 226
    OEDepict::OEBondSVGNoMarkup, 228
    OEDepict::OEDisplayAtomIdx, 229
    OEDepict::OEDisplayAtomMapIdx, 230
    OEDepict::OEDisplayBondIdx, 231
    OEDepict::OEDisplayNoAtomProp, 233
    OEDepict::OEDisplayNoBondProp, 233
    OEDepict::OEFont, 234
    OEDepict::OEHighlightBase, 243
    OEDepict::OEHighlightByBallAndStick, 245
    OEDepict::OEHighlightByCogwheel, 250
    OEDepict::OEHighlightByColor, 258
    OEDepict::OEHighlightByLasso, 262
    OEDepict::OEHighlightByStick, 270
    OEDepict::OEHighlightLabel, 275
    OEDepict::OEHighlightOverlayBase, 280
    OEDepict::OEHighlightOverlayByBallAndStick, 281
    OEDepict::OEImage, 285
    OEDepict::OEImageBase, 295
    OEDepict::OEImageFileBase, 312
    OEDepict::OEImageFileCreatorBase, 313
    OEDepict::OEImageFrame, 315
    OEDepict::OEImageGrid, 324
    OEDepict::OEImageTable, 328
    OEDepict::OEImageTableOptions, 338
    OEDepict::OEJavaImageBase, 355
    OEDepict::OELegendLayout, 356
    OEDepict::OELegendLayoutOptions, 358
    OEDepict::OEMultiPageImageFile, 365
    OEDepict::OENearestAtom, 367
    OEDepict::OENearestBond, 368
    OEDepict::OEPen, 369
    OEDepict::OEPrepareDepictionOptions, 379
    OEDepict::OEReport, 384
    OEDepict::OEReportOptions, 401
    OEDepict::OESVGClass, 413
    OEDepict::OESVGGroup, 414
CreateCopy
    OEDepict::OE2DAtomDisplay, 123
    OEDepict::OE2DBondDisplay, 133
    OEDepict::OE2DMolDisplay, 141
    OEDepict::OEAlignmentResult, 217
    OEDepict::OEAtomSVGAtomIdxMarkup, 220
    OEDepict::OEAtomSVGMarkupBase, 221
    OEDepict::OEAtomSVGNoMarkup, 222
    OEDepict::OEAtomSVGResidueMarkup, 224
    OEDepict::OEBondSVGBondIdxMarkup, 226
    OEDepict::OEBondSVGMarkupBase, 227
    OEDepict::OEBondSVGNoMarkup, 228
    OEDepict::OEDisplayAtomIdx, 229
    OEDepict::OEDisplayAtomMapIdx, 230
    OEDepict::OEDisplayAtomPropBase, 231
    OEDepict::OEDisplayBondIdx, 232
    OEDepict::OEDisplayBondPropBase, 232

OEDepict::OEDisplayNoAtomProp, 233
OEDepict::OEDisplayNoBondProp, 234
OEDepict::OEHighlightBase, 244
OEDepict::OEHighlightByBallAndStick, 245
OEDepict::OEHighlightByCogwheel, 251
OEDepict::OEHighlightByColor, 258
OEDepict::OEHighlightByLasso, 263
OEDepict::OEHighlightByStick, 270
OEDepict::OEImageFileCreatorBase, 313
CreateImage
OEDepict::OEImageFileCreatorBase, 314

D

depict.java
Example Code, 70
depict.txt
Example Code, 71
DrawArc
OEDepict::OEImage, 287
OEDepict::OEImageBase, 295
OEDepict::OEImageFrame, 316
DrawCircle
OEDepict::OEImage, 287
OEDepict::OEImageBase, 297
OEDepict::OEImageFrame, 317
DrawCubicBezier
OEDepict::OEImage, 288
OEDepict::OEImageBase, 298
OEDepict::OEImageFrame, 317
DrawLine
OEDepict::OEImage, 289
OEDepict::OEImageBase, 299
OEDepict::OEImageFrame, 318
DrawPath
OEDepict::OEImage, 289
OEDepict::OEImageBase, 300
OEDepict::OEImageFrame, 318
DrawPie
OEDepict::OEImage, 290
OEDepict::OEImageBase, 301
OEDepict::OEImageFrame, 319
DrawPoint
OEDepict::OEImage, 290
OEDepict::OEImageBase, 302
OEDepict::OEImageFrame, 319
DrawPolygon
OEDepict::OEImage, 291
OEDepict::OEImageBase, 303
OEDepict::OEImageFrame, 320
DrawQuadraticBezier
OEDepict::OEImage, 292
OEDepict::OEImageBase, 304
OEDepict::OEImageFrame, 320
DrawRectangle
OEDepict::OEImage, 292
OEDepict::OEImageBase, 306
OEDepict::OEImageFrame, 321
DrawText
OEDepict::OEImage, 293
OEDepict::OEImageBase, 307
OEDepict::OEImageFrame, 321
OEDepict::OEImageTable, 329
DrawTriangle
OEDepict::OEImage, 293
OEDepict::OEImageBase, 308
OEDepict::OEImageFrame, 322

E

Example Code
depict.java, 70
depict.txt, 71
match2img.java, 74
match2img.txt, 76
mcsalign2d.java, 78
mcsalign2d.txt, 80
mdlquery2img.java, 82
mdlquery2img.txt, 84
mdlreaction2img.java, 86
mdlreaction2img.txt, 88
mol2img.java, 90
mol2img.txt, 92
mols2img.java, 97
mols2img.txt, 98
mols2pdf.java, 103
mols2pdf.txt, 105
ringdict2pdf.java, 107
ringdict2pdf.txt, 108
smartsalign2d.java, 110
smartsalign2d.txt, 111
viewmdlsearch.java, 114
viewmdlsearch.txt, 117

G

GetAddDepictionHydrogens
OEDepict::OEAlignmentOptions, 211
OEDepict::OEPrepareDepictionOptions, 379
GetAlignment
OEDepict::OEFont, 235
GetAntiAliasing
OEDepict::OEJavaImageBase, 356
GetAromaticStyle
OEDepict::OE2DMolDisplayOptions, 149
GetAtom
OEDepict::OEAtomDisplayBase, 218
OEDepict::OENearestAtom, 367
GetAtomColor
OEDepict::OE2DMolDisplayOptions, 150
GetAtomColorStyle

OEDepict::OE2DMolDisplayOptions, 150
GetAtomDisplay
OEDepict::OE2DMolDisplay, 141
GetAtomDisplays
OEDepict::OE2DMolDisplay, 141
GetAtomExternalHighlight
OEDepict::OEHighlightByColor, 258
GetAtomExternalHighlightRatio
OEDepict::OEHighlightByStick, 270
GetAtomLabelFont
OEDepict::OE2DMolDisplayOptions, 150
GetAtomLabelFontScale
OEDepict::OE2DMolDisplayOptions, 150
GetAtomLabelMonochrome
OEDepict::OEHighlightByBallAndStick, 245
OEDepict::OEHighlightByCogwheel, 251
OEDepict::OEHighlightByLasso, 263
OEDepict::OEHighlightByStick, 271
OEDepict::OEHighlightOverlayByBallAndStick, 282
GetAtomPropertyFunctor
OEDepict::OE2DMolDisplayOptions, 151
GetAtomPropLabelFont
OEDepict::OE2DMolDisplayOptions, 151
GetAtomPropLabelFontScale
OEDepict::OE2DMolDisplayOptions, 151
GetAtoms
OEDepict::OEAlignmentResult, 217
GetAtomStereoStyle
OEDepict::OE2DMolDisplayOptions, 152
GetAtomSVGMarkupFunctor
OEDepict::OE2DMolDisplayOptions, 149
GetAtomVisibilityFunctor
OEDepict::OE2DMolDisplayOptions, 151
GetBackColor
OEDepict::OEPen, 369
GetBackgroundColor
OEDepict::OE2DMolDisplayOptions, 152
GetBallRadiusScale
OEDepict::OEHighlightByBallAndStick, 246
OEDepict::OEHighlightByCogwheel, 251
OEDepict::OEHighlightOverlayByBallAndStick, 282
GetBgnPen
OEDepict::OE2DBondDisplay, 133
GetBodies
OEDepict::OEReport, 384
GetBody
OEDepict::OEReport, 385
GetBodyCell
OEDepict::OEImageTable, 329
GetBodyCells
OEDepict::OEImageTable, 330
GetBond
OEDepict::OEBondDisplayBase, 225
OEDepict::OENearestBond, 368
GetBondColorStyle
OEDepict::OE2DMolDisplayOptions, 152
GetBondDisplay
OEDepict::OE2DMolDisplay, 142
GetBondDisplays
OEDepict::OE2DMolDisplay, 142
GetBondLineAtomLabelGapScale
OEDepict::OE2DMolDisplayOptions, 153
GetBondLineGapScale
OEDepict::OE2DMolDisplayOptions, 153
GetBondPropertyFunctor
OEDepict::OE2DMolDisplayOptions, 153
GetBondPropLabelFont
OEDepict::OE2DMolDisplayOptions, 153
GetBondPropLabelFontScale
OEDepict::OE2DMolDisplayOptions, 153
GetBonds
OEDepict::OEAlignmentResult, 217
GetBondStereoStyle
OEDepict::OE2DMolDisplayOptions, 154
GetBondSVGMarkupFunctor
OEDepict::OE2DMolDisplayOptions, 154
GetBondWidthScaling
OEDepict::OE2DMolDisplayOptions, 154
GetBoundingBoxPen
OEDepict::OEHighlightLabel, 277
GetButtonFont
OEDepict::OELegendLayoutOptions, 359
GetButtonHeightScale
OEDepict::OELegendLayoutOptions, 359
GetButtonPen
OEDepict::OELegendLayoutOptions, 359
GetButtonWidthScale
OEDepict::OELegendLayoutOptions, 359
GetCell
OEDepict::OEImageGrid, 324
OEDepict::OEImageTable, 332
OEDepict::OEReport, 386
GetCellBorderPen
OEDepict::OEImageTableOptions, 340
GetCellColor
OEDepict::OEImageTableOptions, 340
GetCellFont
OEDepict::OEImageTableOptions, 340
GetCellGap
OEDepict::OEImageGrid, 324
OEDepict::OEReportOptions, 402
GetCellHeight
OEDepict::OEImageGrid, 325
OEDepict::OEReport, 386
OEDepict::OEReportOptions, 403
GetCells

OEDepict::OEImageGrid, 325
OEDepict::OEImageTable, 333
OEDepict::OEReport, 386
GetCellWidth
OEDepict::OEImageGrid, 325
OEDepict::OEReport, 386
OEDepict::OEReportOptions, 403
GetCharge
OEDepict::OE2DAtomDisplay, 123
GetClassName
OEDepict::OEAtomSVGAtomIdxMarkup, 220
OEDepict::OEAtomSVGMarkupBase, 221
OEDepict::OEAtomSVGNoMarkup, 222
OEDepict::OEAtomSVGResidueMarkup, 224
OEDepict::OEBondSVGBondIdxMarkup, 226
OEDepict::OEBondSVGMarkupBase, 227
OEDepict::OEBondSVGNoMarkup, 228
GetClearCoords
OEDepict::OEAlignmentOptions, 211
OEDepict::OEPrepareDepictionOptions, 380
GetColor
OEDepict::OEFont, 236
OEDepict::OEHighlightByBallAndStick, 246
OEDepict::OEHighlightByCogwheel, 251
OEDepict::OEHighlightByColor, 258
OEDepict::OEHighlightByLasso, 264
OEDepict::OEHighlightByStick, 271
GetColors
OEDepict::OEHighlightOverlayBase, 280
GetColorStyle
OEDepict::OELegendLayoutOptions, 359
GetConsiderAtomLabelBoundingBox
OEDepict::OEHighlightByLasso, 263
GetCoords
OEDepict::OE2DAtomDisplay, 124
GetDataType
OEDepict::OE2DMolDisplay, 143
OEDepict::OEAtomDisplayBase, 218
OEDepict::OEBondDisplayBase, 225
OEDepict::OEMolDisplayBase, 364
GetDefaultBondPen
OEDepict::OE2DMolDisplayOptions, 154
GetDepictOrientation
OEDepict::OEPrepareDepictionOptions, 380
GetDisplayPos
OEDepict::OE2DBondDisplay, 134
GetDisplayPosition
OEDepict::OENearestAtom, 367
OEDepict::OENearestBond, 368
GetDisplayType
OEDepict::OE2DBondDisplay, 134
GetDist
OEDepict::OENearestAtom, 367
OEDepict::OENearestBond, 368
GetEnd
OEDepict::OE2DPath, 205
GetEndPen
OEDepict::OE2DBondDisplay, 135
GetExplicitAtomLabelAngle
OEDepict::OE2DMolDisplayOptions, 155
GetExtension
OEDepict::OEImageFileBase, 312
OEDepict::OEImageFileCreatorBase, 314
GetFamily
OEDepict::OEFont, 236
GetFill
OEDepict::OEPen, 370
GetFixedCoords
OEDepict::OEAlignmentOptions, 211
GetFont
OEDepict::OEHighlightLabel, 277
GetFontScale
OEDepict::OEHighlightLabel, 277
GetFooter
OEDepict::OEReport, 390
GetFooterHeight
OEDepict::OEReport, 390
OEDepict::OEReportOptions, 403
GetFooters
OEDepict::OEReport, 391
GetFooterWidth
OEDepict::OEReport, 391
OEDepict::OEReportOptions, 403
GetForeColor
OEDepict::OEPen, 370
GetGlobalOffset
OEDepict::OEImageBase, 309
GetGraphics
OEDepict::OEJavaImageBase, 356
GetGroupId
OEDepict::OEAtomSVGAtomIdxMarkup, 220
OEDepict::OEAtomSVGMarkupBase, 221
OEDepict::OEAtomSVGNoMarkup, 222
OEDepict::OEAtomSVGResidueMarkup, 224
OEDepict::OEBondSVGBondIdxMarkup, 226
OEDepict::OEBondSVGMarkupBase, 228
OEDepict::OEBondSVGNoMarkup, 229
GetHCount
OEDepict::OE2DAtomDisplay, 124
GetHeader
OEDepict::OEReport, 392
GetHeaderCell
OEDepict::OEImageTable, 334
GetHeaderCells
OEDepict::OEImageTable, 335
GetHeaderColor
OEDepict::OEImageTableOptions, 341
GetHeaderFont

OEDepict::OEImageTableOptions, 341
GetHeaderHeight
OEDepict::OEReport, 392
OEDepict::OEReportOptions, 403
GetHeaders
OEDepict::OEReport, 392
GetHeaderWidth
OEDepict::OEReport, 392
OEDepict::OEReportOptions, 404
GetHeight
OEDepict::OE2DMolDisplay, 143
OEDepict::OE2DMolDisplayOptions, 155
OEDepict::OEImage, 294
OEDepict::OEImageBase, 309
OEDepict::OEImageFrame, 322
GetHPosition
OEDepict::OE2DAtomDisplay, 125
GetHydrogenStyle
OEDepict::OE2DMolDisplayOptions, 155
GetHyperlink
OEDepict::OEFont, 236
GetId
OEDepict::OESVGGroup, 414
GetInnerContour
OEDepict::OEHighlightByCogwheel, 251
GetInteractiveStyle
OEDepict::OELegendLayoutOptions, 360
GetLabel
OEDepict::OE2DAtomDisplay, 125
GetLabelFont
OEDepict::OE2DAtomDisplay, 126
GetLassoScale
OEDepict::OEHighlightByLasso, 263
GetLayer
OEDepict::OE2DMolDisplay, 144
GetLayoutStyle
OEDepict::OELegendLayoutOptions, 360
GetLegendArea
OEDepict::OELegendLayout, 357
GetLegendBoxPen
OEDepict::OELegendLayoutOptions, 360
GetLineCap
OEDepict::OEPen, 370
GetLineJoin
OEDepict::OEPen, 371
GetLineStipple
OEDepict::OEPen, 371
GetLineWidth
OEDepict::OEPen, 371
GetLineWidthScale
OEDepict::OEHighlightByCogwheel, 252
OEDepict::OEHighlightByColor, 259
OEDepict::OEHighlightByLasso, 264
GetMargin
OEDepict::OE2DMolDisplayOptions, 155
OEDepict::OEImageGrid, 326
OEDepict::OEImageTableOptions, 341
OEDepict::OELegendLayoutOptions, 360
GetMass
OEDepict::OE2DAtomDisplay, 126
GetMaxAllowedAtomPairClashes
OEDepict::OEAlignmentOptions, 211
GetMaxBondRotations
OEDepict::OEAlignmentOptions, 212
GetMaxPatternMatches
OEDepict::OEAlignmentOptions, 212
GetMinFontSize
OEDepict::OEImageBase, 309
GetMolecule
OEDepict::OEMolDisplayBase, 364
GetName
OEDepict::OESVGClass, 413
GetOffset
OEDepict::OEImageFrame, 323
GetOptimizeMacrocycles
OEDepict::OEPrepareDepictionOptions, 380
GetOptions
OEDepict::OE2DMolDisplay, 144
OEDepict::OEImageTable, 336
OEDepict::OELegendLayout, 357
OEDepict::OEReport, 393
GetOrientation
OEDepict::OEMultiPageImageFile, 366
GetPage
OEDepict::OEMultiPageImageFile, 366
OEDepict::OEReport, 393
GetPageHeight
OEDepict::OEMultiPageImageFile, 366
OEDepict::OEReport, 394
OEDepict::OEReportOptions, 404
GetPageMargin
OEDepict::OEReportOptions, 404
GetPages
OEDepict::OEMultiPageImageFile, 366
OEDepict::OEReport, 394
GetPageWidth
OEDepict::OEMultiPageImageFile, 366
OEDepict::OEReport, 394
OEDepict::OEReportOptions, 404
GetPen
OEDepict::OEHighlightByLasso, 264
GetPerceiveBondStereo
OEDepict::OEPrepareDepictionOptions, 380
GetPoint
OEDepict::OE2DPathPoint, 207
GetPoints
OEDepict::OE2DPath, 206
GetPointType

OEDepict::OE2DPathPoint, 208
GetPosition
OEDepict::OEHighlightByLasso, 264
GetProperty
OEDepict::OE2DAtomDisplay, 126
OEDepict::OE2DBondDisplay, 135
GetPropertyFont
OEDepict::OE2DAtomDisplay, 127
OEDepict::OE2DBondDisplay, 135
GetPropertyOffset
OEDepict::OE2DAtomDisplay, 127
OEDepict::OE2DBondDisplay, 136
GetProtectiveGroupLabelFont
OEDepict::OE2DMolDisplayOptions, 156
GetProtectiveGroupStyle
OEDepict::OE2DMolDisplayOptions, 156
GetRadical
OEDepict::OE2DAtomDisplay, 128
GetRMSD
OEDepict::OEAlignmentResult, 217
GetRMSDCutoff
OEDepict::OEAlignmentOptions, 212
GetRotateAroundBonds
OEDepict::OEAlignmentOptions, 212
GetRotationAngle
OEDepict::OEFont, 236
GetScale
OEDepict::OE2DMolDisplay, 144
OEDepict::OE2DMolDisplayOptions, 157
GetSize
OEDepict::OEFont, 237
GetStart
OEDepict::OE2DPath, 206
GetStickWidthScale
OEDepict::OEHighlightByBallAndStick, 246
OEDepict::OEHighlightByCogwheel, 252
OEDepict::OEHighlightByStick, 271
OEDepict::OEHighlightOverlayByBallAndStick, 282
GetStippleFactor
OEDepict::OEPen, 371
GetStubColumnCell
OEDepict::OEImageTable, 336
GetStubColumnCells
OEDepict::OEImageTable, 337
GetStubColumnColor
OEDepict::OEImageTableOptions, 341
GetStubColumnFont
OEDepict::OEImageTableOptions, 342
GetStyle
OEDepict::OEFont, 237
GetSuperAtomLabelFont
OEDepict::OE2DMolDisplayOptions, 157
GetSuperAtomStyle
OEDepict::OE2DMolDisplayOptions, 157
GetSuppressHydrogens
OEDepict::OEAlignmentOptions, 212
OEDepict::OEPrepareDepictionOptions, 381
GetSVGClass
OEDepict::OEImageBase, 309
GetSVGGroup
OEDepict::OEImageBase, 309
GetSVGGroups
OEDepict::OEImageBase, 310
GetText
OEDepict::OEHighlightLabel, 278
GetTitleFont
OEDepict::OE2DMolDisplayOptions, 157
GetTitleFontScale
OEDepict::OE2DMolDisplayOptions, 158
GetTitleHeight
OEDepict::OE2DMolDisplayOptions, 158
GetTitleLocation
OEDepict::OE2DMolDisplayOptions, 158
GetWidth
OEDepict::OE2DMolDisplay, 145
OEDepict::OE2DMolDisplayOptions, 158
OEDepict::OEImage, 294
OEDepict::OEImageBase, 309
OEDepict::OEImageFrame, 323
GetX
OEDepict::OE2DPoint, 209
GetY
OEDepict::OE2DPoint, 209

H

HasHeader
OEDepict::OEImageTableOptions, 342
HasHyperlink
OEDepict::OEFont, 237
HasLabel
OEDepict::OE2DAtomDisplay, 128
HasProperty
OEDepict::OE2DAtomDisplay, 129
OEDepict::OE2DBondDisplay, 137
HasProtectiveGroupStyle
OEDepict::OE2DMolDisplayOptions, 159
HasStubColumn
OEDepict::OEImageTableOptions, 342

I

IsBodyPage
OEDepict::OEReport, 394
IsClosed
OEDepict::OE2DPath, 206
IsDataType
OEDepict::OE2DMolDisplay, 145
OEDepict::OEAtomDisplayBase, 219

OEDepict::OEBondDisplayBase, 225
OEDepict::OEMolDisplayBase, 364
IsEmpty
OEDepict::OEImage, 294
IsGridPage
OEDepict::OEReport, 395
IsMultiPage
OEDepict::OEImageFileCreatorBase, 314
IsValid
OEDepict::OE2DMolDisplay, 145
OEDepict::OE2DPath, 207
OEDepict::OEAlignmentResult, 217
OEDepict::OENearestAtom, 367
OEDepict::OENearestBond, 368
IsVisible
OEDepict::OEAtomDisplayBase, 219
OEDepict::OEBondDisplayBase, 225
OEDepict::OEImageFileBase, 312
OEDepict::OESVGGroup, 414

M

match2img.java
Example Code, 74
match2img.txt
Example Code, 76
mcsalign2d.java
Example Code, 78
mcsalign2d.txt
Example Code, 80
mdlquery2img.java
Example Code, 82
mdlquery2img.txt
Example Code, 84
mdlreaction2img.java
Example Code, 86
mdlreaction2img.txt
Example Code, 88
mol2img.java
Example Code, 90
mol2img.txt
Example Code, 92
mols2img.java
Example Code, 97
mols2img.txt
Example Code, 98
mols2pdf.java
Example Code, 103
mols2pdf.txt
Example Code, 105

N

NewBody
OEDepict::OEReport, 396
NewCell
OEDepict::OEReport, 397
NewPage
OEDepict::OEMultiPageImageFile, 366
NewSVGClass
OEDepict::OEImageBase, 310
NewSVGGroup
OEDepict::OEImageBase, 311
NumAtoms
OEDepict::OEAlignmentResult, 217
NumBodyPages
OEDepict::OEReport, 398
NumBonds
OEDepict::OEAlignmentResult, 218
NumCols
OEDepict::OEImageGrid, 326
NumColsPerPage
OEDepict::OEReport, 398
OEDepict::OEReportOptions, 405
NumColumns
OEDepict::OEImageTable, 337
OEDepict::OEImageTableOptions, 342
NumGridPages
OEDepict::OEReport, 398
NumPages
OEDepict::OEMultiPageImageFile, 367
OEDepict::OEReport, 399
NumRows
OEDepict::OEImageGrid, 327
OEDepict::OEImageTable, 337
OEDepict::OEImageTableOptions, 342
NumRowsPerPage
OEDepict::OEReport, 399
OEDepict::OEReportOptions, 405

O

OEDepict::OE2DAtomDisplay, 121
Constructors, 123
CreateCopy, 123
GetCharge, 123
GetCoords, 124
GetHCount, 124
GetHPosition, 125
GetLabel, 125
GetLabelFont, 126
GetMass, 126
GetProperty, 126
GetPropertyFont, 127
GetPropertyOffset, 127
GetRadical, 128
HasLabel, 128
HasProperty, 129
SetCharge, 129
SetCoords, 129
SetHCount, 129

SetHPosition, 129
SetLabel, 130
SetLabelFont, 130
SetMass, 130
SetProperty, 131
SetPropertyFont, 131
SetPropertyOffset, 131
SetRadical, 132
OEDepict::OE2DBondDisplay, 132
Constructors, 133
CreateCopy, 133
GetBgnPen, 133
GetDisplayPos, 134
GetDisplayType, 134
GetEndPen, 135
GetProperty, 135
GetPropertyFont, 135
GetPropertyOffset, 136
HasProperty, 137
SetBgnPen, 137
SetDisplayPos, 137
SetDisplayType, 137
SetEndPen, 138
SetProperty, 138
SetPropertyFont, 138
SetPropertyOffset, 139
OEDepict::OE2DMolDisplay, 139
Constructors, 140
CreateCopy, 141
GetAtomDisplay, 141
GetAtomDisplays, 141
GetBondDisplay, 142
GetBondDisplays, 142
GetDataType, 143
GetHeight, 143
GetLayer, 144
GetOptions, 144
GetScale, 144
GetWidth, 145
IsDataType, 145
IsValid, 145
OEDepict::OE2DMolDisplayOptions, 145
Constructors, 147
GetAromaticStyle, 149
GetAtomColor, 150
GetAtomColorStyle, 150
GetAtomLabelFont, 150
GetAtomLabelFontScale, 150
GetAtomPropertyFunctor, 151
GetAtomPropLabelFont, 151
GetAtomPropLabelFontScale, 151
GetAtomStereoStyle, 152
GetAtomSVGMarkupFunctor, 149
GetAtomVisibilityFunctor, 151
GetBackgroundColor, 152
GetBondColorStyle, 152
GetBondLineAtomLabelGapScale, 153
GetBondLineGapScale, 153
GetBondPropertyFunctor, 153
GetBondPropLabelFont, 153
GetBondPropLabelFontScale, 153
GetBondStereoStyle, 154
GetBondSVGMarkupFunctor, 154
GetBondWidthScaling, 154
GetDefaultBondPen, 154
GetExplicitAtomLabelAngle, 155
GetHeight, 155
GetHydrogenStyle, 155
GetMargin, 155
GetProtectiveGroupLabelFont, 156
GetProtectiveGroupStyle, 156
GetScale, 157
GetSuperAtomLabelFont, 157
GetSuperAtomStyle, 157
GetTitleFont, 157
GetTitleFontScale, 158
GetTitleHeight, 158
GetTitleLocation, 158
GetWidth, 158
HasProtectiveGroupStyle, 159
SetAromaticStyle, 159
SetAtomColor, 160
SetAtomColorStyle, 161
SetAtomLabelFont, 162
SetAtomLabelFontScale, 164
SetAtomPropertyFunctor, 168
SetAtomPropLabelFont, 165
SetAtomPropLabelFontScale, 167
SetAtomStereoStyle, 169
SetAtomSVGMarkupFunctor, 170
SetAtomVisibilityFunctor, 171
SetBackgroundColor, 172
SetBondColorStyle, 173
SetBondLineAtomLabelGapScale, 175
SetBondLineGapScale, 174
SetBondPropertyFunctor, 179
SetBondPropLabelFont, 176
SetBondPropLabelFontScale, 178
SetBondStereoStyle, 180
SetBondSVGMarkupFunctor, 181
SetBondWidthScaling, 182
SetDefaultBondPen, 184
SetDimensions, 185
SetExplicitAtomLabelAngle, 186
SetHeight, 187
SetHydrogenStyle, 188
SetMargin, 189
SetMargins, 190

SetProtectiveGroupLabelFont, 191
SetProtectiveGroupStyle, 192
SetScale, 193
SetSuperAtomLabelFont, 194
SetSuperAtomStyle, 195
SetTitleFont, 195
SetTitleFontScale, 197
SetTitleHeight, 200
SetTitleLocation, 199
SetWidth, 201
OEDepict::OE2DMolDisplaySetup, 414
OEDepict::OE2DMolDisplaySetup::All, 414
OEDepict::OE2DMolDisplaySetup::AromaticStyle, 415
OEDepict::OE2DMolDisplaySetup::AtomColorStyle, 415
OEDepict::OE2DMolDisplaySetup::AtomPropDisplay, 416
OEDepict::OE2DMolDisplaySetup::AtomStereoStyle, 417
OEDepict::OE2DMolDisplaySetup::BondColorStyle, 418
OEDepict::OE2DMolDisplaySetup::BondPropDisplay, 418
OEDepict::OE2DMolDisplaySetup::BondStereoStyle, 419
OEDepict::OE2DMolDisplaySetup::DefaultLineWidth, 419
OEDepict::OE2DMolDisplaySetup::HydrogenStyle, 420
OEDepict::OE2DMolDisplaySetup::ProtectiveGroupStyle, 421
OEDepict::OE2DMolDisplaySetup::Scale, 423
OEDepict::OE2DMolDisplaySetup::SuperAtomStyle, 423
OEDepict::OE2DMolDisplaySetup::TitleLocation, 425
OEDepict::OE2DPath, 202
AddCurveSegment, 204
AddLineSegment, 205
AddStartPoint, 205
Constructors, 203
GetEnd, 205
GetPoints, 206
GetStart, 206
IsClosed, 206
IsValid, 207
SetClosed, 207
OEDepict::OE2DPathPoint, 207
Constructors, 207
GetPoint, 207
GetPointType, 208
OEDepict::OE2DPathPointType, 425
OEDepict::OE2DPathPointType::CurveC1, 425
OEDepict::OE2DPathPointType::CurveC2, 425
OEDepict::OE2DPathPointType::CurveEnd, 426
OEDepict::OE2DPathPointType::LineEnd, 426
OEDepict::OE2DPathPointType::Start, 426
OEDepict::OE2DPoint, 208
Constructors, 208
GetX, 209
GetY, 209
operator+=, 208
SetX, 209
SetY, 209
OEDepict::OEAddBuiltInHiddenSVGClass, 547
OEDepict::OEAddBuiltInVisibleSVGClass, 548
OEDepict::OEAddDepictionHydrogens, 548
OEDepict::OEAddHighlighting, 548
OEDepict::OEAddHighlightOverlay, 556
OEDepict::OEAddInteractiveIcon, 557
OEDepict::OEAddLabel, 557
OEDepict::OEAddOpenEyeIcon, 561
OEDepict::OEAddSVGClickEvent, 562
OEDepict::OEAddSVGHover, 564
OEDepict::OEAddSVGToggle, 565
OEDepict::OEAddWatermark, 566
OEDepict::OEAlignment, 426
OEDepict::OEAlignment::Center, 427
OEDepict::OEAlignment::Default, 427
OEDepict::OEAlignment::Left, 427
OEDepict::OEAlignment::Right, 427
OEDepict::OEAlignmentOptions, 209
Constructors, 210
GetAddDepictionHydrogens, 211
GetClearCoords, 211
GetFixedCoords, 211
GetMaxAllowedAtomPairClashes, 211
GetMaxBondRotations, 212
GetMaxPatternMatches, 212
GetRMSDCutoff, 212
GetRotateAroundBonds, 212
GetSuppressHydrogens, 212
SetAddDepictionHydrogens, 213
SetClearCoords, 213
SetFixedCoords, 213
SetMaxAllowedAtomPairClashes, 214
SetMaxBondRotations, 214
SetMaxPatternMatches, 214
SetRMSDCutoff, 215
SetRotateAroundBonds, 215
SetSuppressHydrogens, 216
OEDepict::OEAlignmentResult, 216
Clear, 217
Constructors, 216
CreateCopy, 217
GetAtoms, 217
GetBonds, 217

GetRMSD, 217
IsValid, 217
NumAtoms, 217
NumBonds, 218
OEDepict::OEAnnotateDepictionProblems, 567
OEDepict::OEAnnotateValenceProblems, 568
OEDepict::OEAromaticStyle, 427
OEDepict::OEAromaticStyle::Circle, 428
OEDepict::OEAromaticStyle::Dash, 429
OEDepict::OEAromaticStyle::Default, 428
OEDepict::OEAromaticStyle::Kekule, 429
OEDepict::OEAtomColorStyle, 430
OEDepict::OEAtomColorStyle::BlackCPK, 430
OEDepict::OEAtomColorStyle::BlackMonochrome, 431
OEDepict::OEAtomColorStyle::Default, 430
OEDepict::OEAtomColorStyle::WhiteCPK, 431
OEDepict::OEAtomColorStyle::WhiteMonochrome, 431
OEDepict::OEAtomDisplayBase, 218
GetAtom, 218
GetDataType, 218
IsDataType, 219
IsVisible, 219
SetVisible, 219
OEDepict::OEAtomStereoStyle, 432
OEDepict::OEAtomStereoStyle::Default, 432
OEDepict::OEAtomStereoStyle::Display, 433
OEDepict::OEAtomStereoStyle::Display::All, 435
OEDepict::OEAtomStereoStyle::Display::AtomStereo, 433
OEDepict::OEAtomStereoStyle::Display::CIPAtomStereo, 434
OEDepict::OEAtomStereoStyle::Display::Default, 433
OEDepict::OEAtomStereoStyle::Display::MDLStereoParity, 434
OEDepict::OEAtomStereoStyle::HashWedgeStyle, 435
OEDepict::OEAtomStereoStyle::HashWedgeStyle::Default, 435
OEDepict::OEAtomStereoStyle::HashWedgeStyle::Hash, 435
OEDepict::OEAtomStereoStyle::HashWedgeStyle::Wedge, 436
OEDepict::OEAtomStereoStyle::Hidden, 432
OEDepict::OEAtomSVGAtomIdxMarkup, 219
Constructors, 219
CreateCopy, 220
GetClassName, 220
GetGroupId, 220
OEDepict::OEAtomSVGMarkupBase, 220
CreateCopy, 221
GetClassName, 221
GetGroupId, 221
OEDepict::OEAtomSVGNoMarkup, 222
Constructors, 222
CreateCopy, 222
GetClassName, 222
GetGroupId, 222
OEDepict::OEAtomSVGResidueMarkup, 222
Constructors, 223
CreateCopy, 224
GetClassName, 224
GetGroupId, 224
OEDepict::OEBondColorStyle, 436
OEDepict::OEBondColorStyle::ByElement, 436
OEDepict::OEBondColorStyle::Default, 436
OEDepict::OEBondColorStyle::Monochrome, 436
OEDepict::OEBondDisplayBase, 224
GetBond, 225
GetDataType, 225
IsDataType, 225
IsVisible, 225
SetVisible, 225
OEDepict::OEBondDisplayPosition, 437
OEDepict::OEBondDisplayPosition::Center, 438
OEDepict::OEBondDisplayPosition::Left, 438
OEDepict::OEBondDisplayPosition::Right, 438
OEDepict::OEBondDisplayType, 439
OEDepict::OEBondDisplayType::Any, 443
OEDepict::OEBondDisplayType::Aromatic, 442
OEDepict::OEBondDisplayType::AromaticDash, 441
OEDepict::OEBondDisplayType::Double, 439
OEDepict::OEBondDisplayType::DoubleArom, 442
OEDepict::OEBondDisplayType::DoubleBowTie, 441
OEDepict::OEBondDisplayType::Hash, 440
OEDepict::OEBondDisplayType::Hidden, 443
OEDepict::OEBondDisplayType::Quadruple, 440
OEDepict::OEBondDisplayType::Single, 439
OEDepict::OEBondDisplayType::SingleArom, 442
OEDepict::OEBondDisplayType::SingleDouble, 441
OEDepict::OEBondDisplayType::Triple, 439
OEDepict::OEBondDisplayType::Wavy, 440
OEDepict::OEBondDisplayType::Wedge, 440
OEDepict::OEBondStereoStyle, 443
OEDepict::OEBondStereoStyle::Default, 443
OEDepict::OEBondStereoStyle::Display, 444
OEDepict::OEBondStereoStyle::Display::All, 445
OEDepict::OEBondStereoStyle::Display::BondStereo, 444
OEDepict::OEBondStereoStyle::Display::CIPBondStereo, 445

OEDepict::OEBondStereoStyle::Display::Default, 444
OEDepict::OEBondStereoStyle::Hidden, 444
OEDepict::OEBondSVGBondIdxMarkup, 225
    Constructors, 226
    CreateCopy, 226
    GetClassName, 226
    GetGroupId, 226
OEDepict::OEBondSVGMarkupBase, 227
    CreateCopy, 227
    GetClassName, 227
    GetGroupId, 228
OEDepict::OEBondSVGNoMarkup, 228
    Constructors, 228
    CreateCopy, 228
    GetClassName, 228
    GetGroupId, 229
OEDepict::OEConfigure2DMolDisplayOptions, 569
OEDepict::OEConfigureColor, 570
OEDepict::OEConfigureHighlightColor, 571
OEDepict::OEConfigureHighlightParams, 572
OEDepict::OEConfigureHighlightStyle, 572
OEDepict::OEConfigureImageGridNumColumns, 573
OEDepict::OEConfigureImageGridNumRows, 573
OEDepict::OEConfigureImageGridParams, 574
OEDepict::OEConfigureImageHeight, 574
OEDepict::OEConfigureImageOptions, 575
OEDepict::OEConfigureImageWidth, 575
OEDepict::OEConfigureMultiPageOrientation, 576
OEDepict::OEConfigureMultiPageParams, 576
OEDepict::OEConfigureMultiPageSize, 577
OEDepict::OEConfigurePrepareDepictionOptions, 577
OEDepict::OEConfigureReportOptions, 578
OEDepict::OECreateWin32GraphicsImage, 578
OEDepict::OEDepict, 446
OEDepict::OEDepict::OEBlackBoxPen, 446
OEDepict::OEDepict::OEBlackPen, 446
OEDepict::OEDepict::OEBlueBoxPen, 446
OEDepict::OEDepict::OEBluePen, 447
OEDepict::OEDepict::OEDarkBackgroundColor, 446
OEDepict::OEDepict::OEDefaultFont, 447
OEDepict::OEDepict::OEGreenBoxPen, 447
OEDepict::OEDepict::OEGreenPen, 447
OEDepict::OEDepict::OELightBackgroundColor, 446
OEDepict::OEDepict::OELightGreyBoxPen, 447
OEDepict::OEDepict::OELightGreyPen, 448
OEDepict::OEDepict::OERedBoxPen, 448
OEDepict::OEDepict::OERedPen, 448
OEDepict::OEDepict::OESVGAreaPen, 448
OEDepict::OEDepict::OETransparentPen, 448
OEDepict::OEDepict::OEWhiteBoxPen, 449
OEDepict::OEDepict::OEWhitePen, 449
OEDepict::OEDepictCoordinates, 578
OEDepict::OEDepictGetArch, 579
OEDepict::OEDepictGetLicensee, 579
OEDepict::OEDepictGetPlatform, 579
OEDepict::OEDepictGetRelease, 579
OEDepict::OEDepictGetSite, 579
OEDepict::OEDepictGetVersion, 579
OEDepict::OEDepictIsLicensed, 579
OEDepict::OEDepictOrientation, 449
OEDepict::OEDepictOrientation::Default, 449
OEDepict::OEDepictOrientation::Horizontal, 449
OEDepict::OEDepictOrientation::Original, 450
OEDepict::OEDepictOrientation::Square, 450
OEDepict::OEDepictOrientation::Vertical, 451
OEDepict::OEDisplayAtomIdx, 229
    Constructors, 229
    CreateCopy, 229
    operator(), 229
OEDepict::OEDisplayAtomMapIdx, 229
    Constructors, 230
    CreateCopy, 230
    operator(), 230
OEDepict::OEDisplayAtomPropBase, 230
    CreateCopy, 231
    operator(), 231
OEDepict::OEDisplayBondIdx, 231
    Constructors, 231
    CreateCopy, 232
    operator(), 231
OEDepict::OEDisplayBondPropBase, 232
    CreateCopy, 232
    operator(), 232
OEDepict::OEDisplayNoAtomProp, 232
    Constructors, 233
    CreateCopy, 233
    operator(), 233
OEDepict::OEDisplayNoBondProp, 233
    Constructors, 233
    CreateCopy, 234
    operator(), 234
OEDepict::OEDrawBorder, 579
OEDepict::OEDrawChevronArrow, 580
OEDepict::OEDrawCurvedArrow, 581
OEDepict::OEDrawCurvedBorder, 582
OEDepict::OEDrawCurvedRectangle, 582
OEDepict::OEDrawEquilibriumArrow, 583
OEDepict::OEDrawHighlighting, 584
OEDepict::OEDrawLegendLayout, 584
OEDepict::OEDrawReactionArrow, 585
OEDepict::OEDrawResonanceArrow, 586
OEDepict::OEDrawSVGHoverText, 586

OEDepict::OEDrawSVGToggleText, 588
OEDepict::OEDrawText, 590
OEDepict::OEDrawTextToCenter, 590
OEDepict::OEEstimateTextWidth, 591
OEDepict::OEFill, 451
OEDepict::OEFill::Default, 452
OEDepict::OEFill::Off, 452
OEDepict::OEFill::On, 452
OEDepict::OEFont, 234
    Constructors, 234
    GetAlignment, 235
    GetColor, 236
    GetFamily, 236
    GetHyperlink, 236
    GetRotationAngle, 236
    GetSize, 237
    GetStyle, 237
    HasHyperlink, 237
    SetAlignment, 237
    SetColor, 238
    SetFamily, 239
    SetHyperlink, 239
    SetRotationAngle, 241
    SetSize, 242
    SetStyle, 243
OEDepict::OEFontFamily, 452
OEDepict::OEFontFamily::Arial, 453
OEDepict::OEFontFamily::Courier, 453
OEDepict::OEFontFamily::Default, 453
OEDepict::OEFontFamily::Helvetica, 453
OEDepict::OEFontFamily::Times, 453
OEDepict::OEFontStyle, 454
OEDepict::OEFontStyle::Bold, 454
OEDepict::OEFontStyle::Default, 454
OEDepict::OEFontStyle::Italic, 454
OEDepict::OEFontStyle::Normal, 454
OEDepict::OEGetAverageBondLengthInDispCoords, 591
OEDepict::OEGetBuiltInHiddenSVGClass, 591
OEDepict::OEGetBuiltInVisibleSVGClass, 592
OEDepict::OEGetCenter, 592
OEDepict::OEGetColor, 592
OEDepict::OEGetDefaultAtomColor, 593
OEDepict::OEGetDisplayCoords, 593
OEDepict::OEGetHighlightColor, 593
OEDepict::OEGetHighlightStyle, 594
OEDepict::OEGetImageGridNumColumns, 594
OEDepict::OEGetImageGridNumRows, 595
OEDepict::OEGetImageHeight, 595
OEDepict::OEGetImageWidth, 595
OEDepict::OEGetLabelPosition, 596
OEDepict::OEGetMoleculeCoords, 597
OEDepict::OEGetMoleculeScale, 597
OEDepict::OEGetMultiPageOrientation, 597
OEDepict::OEGetMultiPageSize, 598
OEDepict::OEGetNearbyAtom, 598
OEDepict::OEGetNearbyBond, 599
OEDepict::OEGetNearestAtom, 599
OEDepict::OEGetNearestBond, 600
OEDepict::OEGetPageHeight, 601
OEDepict::OEGetPageWidth, 601
OEDepict::OEGetSupportedImageFileExtensions, 601
OEDepict::OEGetSupportedMultiPageImageFileExtensions, 602
OEDepict::OEHasDepictionHydrogens, 602
OEDepict::OEHighlightBase, 243
    Constructors, 243
    CreateCopy, 244
OEDepict::OEHighlightByBallAndStick, 244
    Constructors, 245
    CreateCopy, 245
    GetAtomLabelMonochrome, 245
    GetBallRadiusScale, 246
    GetColor, 246
    GetStickWidthScale, 246
    SetAtomLabelMonochrome, 246
    SetBallRadiusScale, 247
    SetColor, 248
    SetStickWidthScale, 248
OEDepict::OEHighlightByCogwheel, 249
    Constructors, 250
    CreateCopy, 251
    GetAtomLabelMonochrome, 251
    GetBallRadiusScale, 251
    GetColor, 251
    GetInnerContour, 251
    GetLineWidthScale, 252
    GetStickWidthScale, 252
    SetAtomLabelMonochrome, 252
    SetBallRadiusScale, 253
    SetColor, 254
    SetInnerContour, 254
    SetLineWidthScale, 255
    SetStickWidthScale, 256
OEDepict::OEHighlightByColor, 257
    Constructors, 258
    CreateCopy, 258
    GetAtomExternalHighlight, 258
    GetColor, 258
    GetLineWidthScale, 259
    SetAtomExternalHighlight, 259
    SetColor, 260
    SetLineWidthScale, 260
OEDepict::OEHighlightByLasso, 261
    Constructors, 262
    CreateCopy, 263
    GetAtomLabelMonochrome, 263

GetColor, 264
GetConsiderAtomLabelBoundingBox, 263
GetLassoScale, 263
GetLineWidthScale, 264
GetPen, 264
GetPosition, 264
SetAtomLabelMonochrome, 265
SetColor, 267
SetConsiderAtomLabelBoundingBox, 265
SetLassoScale, 266
SetLineWidthScale, 267
SetPen, 268
SetPosition, 269
OEDepict::OEHighlightByStick, 269
Constructors, 270
CreateCopy, 270
GetAtomExternalHighlightRatio, 270
GetAtomLabelMonochrome, 271
GetColor, 271
GetStickWidthScale, 271
SetAtomExternalHighlightRatio, 271
SetAtomLabelMonochrome, 272
SetColor, 273
SetStickWidthScale, 274
OEDepict::OEHighlightLabel, 274
Constructors, 275
GetBoundingBoxPen, 277
GetFont, 277
GetFontScale, 277
GetText, 278
SetBoundingBoxPen, 278
SetFont, 278
SetFontScale, 279
OEDepict::OEHighlightOverlayBase, 280
Constructors, 280
GetColors, 280
SetColors, 280
OEDepict::OEHighlightOverlayByBallAndStick, 280
Constructors, 281
GetAtomLabelMonochrome, 282
GetBallRadiusScale, 282
GetStickWidthScale, 282
SetAtomLabelMonochrome, 283
SetBallRadiusScale, 283
SetStickWidthScale, 284
OEDepict::OEHighlightOverlayStyle, 455
OEDepict::OEHighlightOverlayStyle::BallAndStick, 455
OEDepict::OEHighlightOverlayStyle::Default, 455
OEDepict::OEHighlightParamName, 456
OEDepict::OEHighlightParamName::Color, 457
OEDepict::OEHighlightParamName::Style, 457
OEDepict::OEHighlightSetup, 457
OEDepict::OEHighlightSetup::All, 458
OEDepict::OEHighlightSetup::Color, 458
OEDepict::OEHighlightSetup::Style, 458
OEDepict::OEHighlightStyle, 459
OEDepict::OEHighlightStyle::BallAndStick, 459
OEDepict::OEHighlightStyle::Cogwheel, 462
OEDepict::OEHighlightStyle::Color, 461
OEDepict::OEHighlightStyle::Default, 459
OEDepict::OEHighlightStyle::Lasso, 462
OEDepict::OEHighlightStyle::Stick, 464
OEDepict::OEHydrogenPos, 465
OEDepict::OEHydrogenPos::East, 466
OEDepict::OEHydrogenPos::North, 466
OEDepict::OEHydrogenPos::South, 466
OEDepict::OEHydrogenPos::Undefined, 465
OEDepict::OEHydrogenPos::West, 466
OEDepict::OEHydrogenStyle, 466
OEDepict::OEHydrogenStyle::Default, 467
OEDepict::OEHydrogenStyle::ExplicitAll, 467
OEDepict::OEHydrogenStyle::ExplicitHetero, 467
OEDepict::OEHydrogenStyle::ExplicitTerminal, 468
OEDepict::OEHydrogenStyle::Hidden, 468
OEDepict::OEHydrogenStyle::ImplicitAll, 469
OEDepict::OEHydrogenStyle::ImplicitHetero, 469
OEDepict::OEHydrogenStyle::ImplicitTerminal, 470
OEDepict::OEIconLocation, 470
OEDepict::OEIconLocation::BottomLeft, 470
OEDepict::OEIconLocation::BottomRight, 471
OEDepict::OEIconLocation::Default, 470
OEDepict::OEIconLocation::TopLeft, 471
OEDepict::OEIconLocation::TopRight, 471
OEDepict::OEImage, 285
Clear, 286
Constructors, 285
DrawArc, 287
DrawCircle, 287
DrawCubicBezier, 288
DrawLine, 289
DrawPath, 289
DrawPie, 290
DrawPoint, 290
DrawPolygon, 291
DrawQuadraticBezier, 292
DrawRectangle, 292
DrawText, 293
DrawTriangle, 293
GetHeight, 294
GetWidth, 294
IsEmpty, 294

Render, 294
OEDepict::OEImageBase, 294
Clear, 295
Constructors, 295
DrawArc, 295
DrawCircle, 297
DrawCubicBezier, 298
DrawLine, 299
DrawPath, 300
DrawPie, 301
DrawPoint, 302
DrawPolygon, 303
DrawQuadraticBezier, 304
DrawRectangle, 306
DrawText, 307
DrawTriangle, 308
GetGlobalOffset, 309
GetHeight, 309
GetMinFontSize, 309
GetSVGClass, 309
GetSVGGroup, 309
GetSVGGroups, 310
GetWidth, 309
NewSVGClass, 310
NewSVGGroup, 311
PopGroup, 311
PushGroup, 311
SetMinFontSize, 309
OEDepict::OEImageFileBase, 312
Constructors, 312
GetExtension, 312
IsVisible, 312
PopGroup, 312
PushGroup, 313
Write, 313
OEDepict::OEImageFileCreatorBase, 313
Constructors, 313
CreateCopy, 313
CreateImage, 314
GetExtension, 314
IsMultiPage, 314
OEDepict::OEImageFrame, 314
Clear, 316
Constructors, 315
DrawArc, 316
DrawCircle, 317
DrawCubicBezier, 317
DrawLine, 318
DrawPath, 318
DrawPie, 319
DrawPoint, 319
DrawPolygon, 320
DrawQuadraticBezier, 320
DrawRectangle, 321
DrawText, 321
DrawTriangle, 322
GetHeight, 322
GetOffset, 323
GetWidth, 323
OEDepict::OEImageGrid, 323
Constructors, 324
GetCell, 324
GetCellGap, 324
GetCellHeight, 325
GetCells, 325
GetCellWidth, 325
GetMargin, 326
NumCols, 326
NumRows, 327
SetCellGap, 327
SetMargin, 327
SetMargins, 327
OEDepict::OEImageGridParamName, 471
OEDepict::OEImageGridParamName::NumColumns, 472
OEDepict::OEImageGridParamName::NumRows, 472
OEDepict::OEImageGridSetup, 472
OEDepict::OEImageGridSetup::All, 472
OEDepict::OEImageGridSetup::NumColumns, 472
OEDepict::OEImageGridSetup::NumRows, 473
OEDepict::OEImageParamName, 474
OEDepict::OEImageParamName::Height, 474
OEDepict::OEImageParamName::Width, 474
OEDepict::OEImageSetup, 474
OEDepict::OEImageSetup::All, 475
OEDepict::OEImageSetup::Height, 475
OEDepict::OEImageSetup::Width, 475
OEDepict::OEImageTable, 328
Constructors, 328
DrawText, 329
GetBodyCell, 329
GetBodyCells, 330
GetCell, 332
GetCells, 333
GetHeaderCell, 334
GetHeaderCells, 335
GetOptions, 336
GetStubColumnCell, 336
GetStubColumnCells, 337
NumColumns, 337
NumRows, 337
OEDepict::OEImageTableOptions, 338
Constructors, 338
GetCellBorderPen, 340
GetCellColor, 340
GetCellFont, 340
GetHeaderColor, 341
GetHeaderFont, 341

GetMargin, 341
GetStubColumnColor, 341
GetStubColumnFont, 342
HasHeader, 342
HasStubColumn, 342
NumColumns, 342
NumRows, 342
SetBaseFontSize, 343
SetCellBorderPen, 343
SetCellColor, 344
SetCellFont, 344
SetColumnWidths, 345
SetHeader, 346
SetHeaderColor, 347
SetHeaderFont, 348
SetMargin, 349
SetMargins, 350
SetRowHeights, 351
SetStubColumn, 352
SetStubColumnColor, 353
SetStubColumnFont, 354
OEDepict::OEImageTableStyle, 476
OEDepict::OEImageTableStyle::Default, 476
OEDepict::OEImageTableStyle::LightBlue, 476
OEDepict::OEImageTableStyle::LightGreen, 476
OEDepict::OEImageTableStyle::LightGrey, 477
OEDepict::OEImageTableStyle::MediumBlue, 477
OEDepict::OEImageTableStyle::MediumGreen, 477
OEDepict::OEImageTableStyle::MediumGrey, 478
OEDepict::OEImageTableStyle::NoStyle, 478
OEDepict::OEImageTableStyle::SantaFe, 478
OEDepict::OEInteractiveEffect, 479
OEDepict::OEInteractiveEffect::Hover, 479
OEDepict::OEInteractiveEffect::None, 479
OEDepict::OEInteractiveEffect::Toggle, 479
OEDepict::OEIsRegisteredImageFile, 602
OEDepict::OEIsRegisteredMultiPageImageFile, 603
OEDepict::OEIsValidSVGClassName, 603
OEDepict::OEIsValidSVGGroupId, 603
OEDepict::OEJavaImageBase, 355
Constructors, 355
GetAntiAliasing, 356
GetGraphics, 356
SetAntiAliasing, 356
OEDepict::OELayerPosition, 479
OEDepict::OELayerPosition::Above, 479
OEDepict::OELayerPosition::Below, 480
OEDepict::OELayerPosition::SVG, 480
OEDepict::OELegendColorStyle, 480
OEDepict::OELegendColorStyle::Default, 480
OEDepict::OELegendColorStyle::LightBlue, 481
OEDepict::OELegendColorStyle::LightGreen, 481
OEDepict::OELegendColorStyle::LightGrey, 481
OEDepict::OELegendColorStyle::NoStyle, 482
OEDepict::OELegendInteractiveStyle, 482
OEDepict::OELegendInteractiveStyle::Default, 482
OEDepict::OELegendInteractiveStyle::Hover, 482
OEDepict::OELegendInteractiveStyle::Permanent, 483
OEDepict::OELegendInteractiveStyle::Toggle, 483
OEDepict::OELegendLayout, 356
Constructors, 356
GetLegendArea, 357
GetOptions, 357
OEDepict::OELegendLayoutOptions, 358
Constructors, 358
GetButtonFont, 359
GetButtonHeightScale, 359
GetButtonPen, 359
GetButtonWidthScale, 359
GetColorStyle, 359
GetInteractiveStyle, 360
GetLayoutStyle, 360
GetLegendBoxPen, 360
GetMargin, 360
SetButtonFont, 360
SetButtonHeightScale, 361
SetButtonPen, 361
SetButtonWidthScale, 361
SetLegendBoxPen, 361
SetMargin, 361
SetMargins, 362
OEDepict::OELegendLayoutStyle, 483
OEDepict::OELegendLayoutStyle::Default, 483
OEDepict::OELegendLayoutStyle::HorizontalBottomLeft, 483
OEDepict::OELegendLayoutStyle::HorizontalBottomRight, 484
OEDepict::OELegendLayoutStyle::HorizontalTopLeft, 484
OEDepict::OELegendLayoutStyle::HorizontalTopRight, 484
OEDepict::OELegendLayoutStyle::VerticalBottomLeft, 485
OEDepict::OELegendLayoutStyle::VerticalBottomRight, 485
OEDepict::OELegendLayoutStyle::VerticalTopLeft, 485
OEDepict::OELegendLayoutStyle::VerticalTopRight, 486
OEDepict::OELengthenVector, 603
OEDepict::OELineCap, 486
OEDepict::OELineCap::Butt, 486
OEDepict::OELineCap::Default, 486

OEDepict::OELineCap::Round, 487
OEDepict::OELineCap::Square, 487
OEDepict::OELineJoin, 487
OEDepict::OELineJoin::Bevel, 488
OEDepict::OELineJoin::Default, 487
OEDepict::OELineJoin::Miter, 488
OEDepict::OELineJoin::Round, 488
OEDepict::OEMargin, 488
OEDepict::OEMargin::Bottom, 489
OEDepict::OEMargin::Left, 489
OEDepict::OEMargin::Right, 489
OEDepict::OEMargin::Top, 489
OEDepict::OEMolDisplayBase, 363
    GetDataType, 364
    GetMolecule, 364
    IsDataType, 364
OEDepict::OEMultiPageImageFile, 364
    Constructors, 365
    GetOrientation, 366
    GetPage, 366
    GetPageHeight, 366
    GetPages, 366
    GetPageWidth, 366
    NewPage, 366
    NumPages, 367
OEDepict::OEMultiPageParamName, 489
OEDepict::OEMultiPageParamName::PageOrientation, 489
OEDepict::OEMultiPageParamName::PageSize, 489
OEDepict::OEMultiPageSetup, 490
OEDepict::OEMultiPageSetup::All, 490
OEDepict::OEMultiPageSetup::PageOrientation, 490
OEDepict::OEMultiPageSetup::PageSize, 491
OEDepict::OENearestAtom, 367
    Constructors, 367
    GetAtom, 367
    GetDisplayPosition, 367
    GetDist, 367
    IsValid, 367
OEDepict::OENearestBond, 367
    Constructors, 368
    GetBond, 368
    GetDisplayPosition, 368
    GetDist, 368
    IsValid, 368
OEDepict::OEOffsetMolDisplay, 604
OEDepict::OEPageOrientation, 491
OEDepict::OEPageOrientation::Default, 491
OEDepict::OEPageOrientation::Landscape, 491
OEDepict::OEPageOrientation::Portrait, 491
OEDepict::OEPageSize, 491
OEDepict::OEPageSize::Custom, 492
OEDepict::OEPageSize::Default, 492
OEDepict::OEPageSize::ISO_A4, 492
OEDepict::OEPageSize::US_Legal, 492
OEDepict::OEPageSize::US_Letter, 492
OEDepict::OEPen, 368
    Constructors, 369
    GetBackColor, 369
    GetFill, 370
    GetForeColor, 370
    GetLineCap, 370
    GetLineJoin, 371
    GetLineStipple, 371
    GetLineWidth, 371
    GetStippleFactor, 371
    SetBackColor, 372
    SetFill, 373
    SetForeColor, 373
    SetLineCap, 373
    SetLineJoin, 374
    SetLineStipple, 375
    SetLineWidth, 375
    SetStippleFactor, 376
OEDepict::OEPrepareAlignedDepiction, 604
OEDepict::OEPrepareDepiction, 609
OEDepict::OEPrepareDepictionOptions, 376
    Constructors, 379
    GetAddDepictionHydrogens, 379
    GetClearCoords, 380
    GetDepictOrientation, 380
    GetOptimizeMacrocycles, 380
    GetPerceiveBondStereo, 380
    GetSuppressHydrogens, 381
    SetAddDepictionHydrogens, 381
    SetClearCoords, 381
    SetDepictOrientation, 381
    SetOptimizeMacrocycles, 382
    SetPerceiveBondStereo, 382
    SetSuppressHydrogens, 382
OEDepict::OEPrepareDepictionSetup, 492
OEDepict::OEPrepareDepictionSetup::All, 492
OEDepict::OEPrepareDepictionSetup::ClearCoords, 493
OEDepict::OEPrepareDepictionSetup::DepictOrientation, 493
OEDepict::OEPrepareDepictionSetup::SuppressHydrogens, 493
OEDepict::OEPrepareMultiAlignedDepiction, 610
OEDepict::OEProtectiveGroupStyle, 494
OEDepict::OEProtectiveGroupStyle::Ac, 495
OEDepict::OEProtectiveGroupStyle::Ad, 495
OEDepict::OEProtectiveGroupStyle::All, 494
OEDepict::OEProtectiveGroupStyle::Alloc, 495
OEDepict::OEProtectiveGroupStyle::Benzoyl, 499
OEDepict::OEProtectiveGroupStyle::Benzyl, 496

OEDepict::OEProtectiveGroupStyle::Bn, 496
OEDepict::OEProtectiveGroupStyle::Boc, 496
OEDepict::OEProtectiveGroupStyle::Bom, 497
OEDepict::OEProtectiveGroupStyle::Bpoc, 497
OEDepict::OEProtectiveGroupStyle::Bs, 497
OEDepict::OEProtectiveGroupStyle::Bt, 498
OEDepict::OEProtectiveGroupStyle::Btm, 498
OEDepict::OEProtectiveGroupStyle::Bz, 498
OEDepict::OEProtectiveGroupStyle::Bzh, 499
OEDepict::OEProtectiveGroupStyle::Bzl, 496
OEDepict::OEProtectiveGroupStyle::BzOM, 499
OEDepict::OEProtectiveGroupStyle::Cbz, 527
OEDepict::OEProtectiveGroupStyle::cHx, 500
OEDepict::OEProtectiveGroupStyle::Dan, 500
OEDepict::OEProtectiveGroupStyle::Dansyl, 500
OEDepict::OEProtectiveGroupStyle::DCB, 501
OEDepict::OEProtectiveGroupStyle::DDE, 501
OEDepict::OEProtectiveGroupStyle::DDIV, 501
OEDepict::OEProtectiveGroupStyle::Default, 494
OEDepict::OEProtectiveGroupStyle::DEIPS, 502
OEDepict::OEProtectiveGroupStyle::DMAB, 502
OEDepict::OEProtectiveGroupStyle::DMIPS, 502
OEDepict::OEProtectiveGroupStyle::DMPM, 503
OEDepict::OEProtectiveGroupStyle::DMPS, 503
OEDepict::OEProtectiveGroupStyle::DMTr, 503
OEDepict::OEProtectiveGroupStyle::DNBZ, 504
OEDepict::OEProtectiveGroupStyle::DNP, 504
OEDepict::OEProtectiveGroupStyle::DOC, 504
OEDepict::OEProtectiveGroupStyle::DPIPS, 505
OEDepict::OEProtectiveGroupStyle::Dpp, 505
OEDepict::OEProtectiveGroupStyle::DTBMS, 505
OEDepict::OEProtectiveGroupStyle::EE, 506
OEDepict::OEProtectiveGroupStyle::ESC, 506
OEDepict::OEProtectiveGroupStyle::Fmoc, 506
OEDepict::OEProtectiveGroupStyle::FourOMeBz, 507
OEDepict::OEProtectiveGroupStyle::FPMP, 507
OEDepict::OEProtectiveGroupStyle::Im, 507
OEDepict::OEProtectiveGroupStyle::Lev, 508
OEDepict::OEProtectiveGroupStyle::MDIPS, 508
OEDepict::OEProtectiveGroupStyle::MDPS, 508
OEDepict::OEProtectiveGroupStyle::MEM, 509
OEDepict::OEProtectiveGroupStyle::Mes, 509
OEDepict::OEProtectiveGroupStyle::MMTR, 509
OEDepict::OEProtectiveGroupStyle::MOM, 510
OEDepict::OEProtectiveGroupStyle::Moz, 510
OEDepict::OEProtectiveGroupStyle::MPC, 510
OEDepict::OEProtectiveGroupStyle::MPM, 511
OEDepict::OEProtectiveGroupStyle::MS, 511
OEDepict::OEProtectiveGroupStyle::MTHP, 511
OEDepict::OEProtectiveGroupStyle::MTM, 512
OEDepict::OEProtectiveGroupStyle::MTr, 512
OEDepict::OEProtectiveGroupStyle::Np, 512
OEDepict::OEProtectiveGroupStyle::NPE, 513
OEDepict::OEProtectiveGroupStyle::NPEOC, 513
OEDepict::OEProtectiveGroupStyle::NPES, 513
OEDepict::OEProtectiveGroupStyle::NPS, 514
OEDepict::OEProtectiveGroupStyle::NVOC, 514
OEDepict::OEProtectiveGroupStyle::OBzl, 514
OEDepict::OEProtectiveGroupStyle::Off, 494
OEDepict::OEProtectiveGroupStyle::ONOS, 515
OEDepict::OEProtectiveGroupStyle::OtBu, 515
OEDepict::OEProtectiveGroupStyle::PAC, 515
OEDepict::OEProtectiveGroupStyle::PBF, 516
OEDepict::OEProtectiveGroupStyle::PCB, 516
OEDepict::OEProtectiveGroupStyle::PHF, 516
OEDepict::OEProtectiveGroupStyle::Pht, 517
OEDepict::OEProtectiveGroupStyle::Piv, 517
OEDepict::OEProtectiveGroupStyle::PMB, 517
OEDepict::OEProtectiveGroupStyle::PMBM, 518
OEDepict::OEProtectiveGroupStyle::Poc, 518
OEDepict::OEProtectiveGroupStyle::PPi, 518
OEDepict::OEProtectiveGroupStyle::SEM, 519
OEDepict::OEProtectiveGroupStyle::SES, 519
OEDepict::OEProtectiveGroupStyle::StBu, 519
OEDepict::OEProtectiveGroupStyle::TBDMS, 520
OEDepict::OEProtectiveGroupStyle::TBDPS, 520
OEDepict::OEProtectiveGroupStyle::TBMPS, 520
OEDepict::OEProtectiveGroupStyle::TBS, 521
OEDepict::OEProtectiveGroupStyle::TCP, 521
OEDepict::OEProtectiveGroupStyle::Teoc, 521
OEDepict::OEProtectiveGroupStyle::TES, 522
OEDepict::OEProtectiveGroupStyle::Tf, 522
OEDepict::OEProtectiveGroupStyle::TFA, 522
OEDepict::OEProtectiveGroupStyle::Thexyl, 523
OEDepict::OEProtectiveGroupStyle::THF, 523
OEDepict::OEProtectiveGroupStyle::THP, 523
OEDepict::OEProtectiveGroupStyle::TIPS, 524
OEDepict::OEProtectiveGroupStyle::TMOB, 524
OEDepict::OEProtectiveGroupStyle::TMS, 524
OEDepict::OEProtectiveGroupStyle::Tos, 526
OEDepict::OEProtectiveGroupStyle::transCinnamyl, 525
OEDepict::OEProtectiveGroupStyle::Trisyl, 525
OEDepict::OEProtectiveGroupStyle::Troc, 525
OEDepict::OEProtectiveGroupStyle::TRT, 526
OEDepict::OEProtectiveGroupStyle::TS, 526
OEDepict::OEProtectiveGroupStyle::Xyl, 527
OEDepict::OEProtectiveGroupStyle::Z, 527
OEDepict::OERadicalDisplayType, 527
OEDepict::OERadicalDisplayType::Diradical, 528
OEDepict::OERadicalDisplayType::Monoradical, 528
OEDepict::OERadicalDisplayType::Off, 527
OEDepict::OEReduceFontSizeToFit, 612
OEDepict::OERegisterImageFile, 613

OEDepict::OERenderMolecule, 613
OEDepict::OERenderMoleculeToString, 617
OEDepict::OEReport, 383
    Constructors, 384
    GetBodies, 384
    GetBody, 385
    GetCell, 386
    GetCellHeight, 386
    GetCells, 386
    GetCellWidth, 386
    GetFooter, 390
    GetFooterHeight, 390
    GetFooters, 391
    GetFooterWidth, 391
    GetHeader, 392
    GetHeaderHeight, 392
    GetHeaders, 392
    GetHeaderWidth, 392
    GetOptions, 393
    GetPage, 393
    GetPageHeight, 394
    GetPages, 394
    GetPageWidth, 394
    IsBodyPage, 394
    IsGridPage, 395
    NewBody, 396
    NewCell, 397
    NumBodyPages, 398
    NumColsPerPage, 398
    NumGridPages, 398
    NumPages, 399
    NumRowsPerPage, 399
OEDepict::OEReportOptions, 400
    Constructors, 401
    GetCellGap, 402
    GetCellHeight, 403
    GetCellWidth, 403
    GetFooterHeight, 403
    GetFooterWidth, 403
    GetHeaderHeight, 403
    GetHeaderWidth, 404
    GetPageHeight, 404
    GetPageMargin, 404
    GetPageWidth, 404
    NumColsPerPage, 405
    NumRowsPerPage, 405
    SetCellGap, 405
    SetFooterHeight, 406
    SetHeaderHeight, 407
    SetPageHeight, 408
    SetPageMargin, 409
    SetPageMargins, 409
    SetPageOrientation, 410
    SetPageSize, 411
    SetPageWidth, 412
OEDepict::OEReportSetup, 528
OEDepict::OEReportSetup::All, 528
OEDepict::OEReportSetup::NumColsPerPage, 529
OEDepict::OEReportSetup::NumRowsPerPage, 529
OEDepict::OEReportSetup::PageHeight, 530
OEDepict::OEReportSetup::PageOrientation, 530
OEDepict::OEReportSetup::PageSize, 530
OEDepict::OEReportSetup::PageWidth, 531
OEDepict::OERotateVector, 618
OEDepict::OEScale, 531
OEDepict::OEScale::AutoScale, 532
OEDepict::OEScale::Default, 532
OEDepict::OESetup2DMolDisplayOptions, 618
OEDepict::OESetupPrepareDepictionOptions, 618
OEDepict::OESetupReportOptions, 619
OEDepict::OEStipple, 532
OEDepict::OEStipple::Default, 532
OEDepict::OEStipple::Dot, 532
OEDepict::OEStipple::DotDash, 533
OEDepict::OEStipple::DotDashDash, 533
OEDepict::OEStipple::LongDash, 533
OEDepict::OEStipple::NoLine, 533
OEDepict::OEStipple::ShortDash, 534
OEDepict::OEStipple::Solid, 534
OEDepict::OESuperAtomStyle, 534
OEDepict::OESuperAtomStyle::All, 535
OEDepict::OESuperAtomStyle::Bu, 541
OEDepict::OESuperAtomStyle::Carbon, 545
OEDepict::OESuperAtomStyle::CCl3, 540
OEDepict::OESuperAtomStyle::CF3, 540
OEDepict::OESuperAtomStyle::CHO, 536
OEDepict::OESuperAtomStyle::CN, 536
OEDepict::OESuperAtomStyle::CNO, 538
OEDepict::OESuperAtomStyle::COO_, 535
OEDepict::OESuperAtomStyle::COOH, 535
OEDepict::OESuperAtomStyle::Default, 534
OEDepict::OESuperAtomStyle::Et, 541
OEDepict::OESuperAtomStyle::Ether, 545
OEDepict::OESuperAtomStyle::Halide, 545
OEDepict::OESuperAtomStyle::iBu, 542
OEDepict::OESuperAtomStyle::Me, 540
OEDepict::OESuperAtomStyle::N2, 537
OEDepict::OESuperAtomStyle::N3, 537
OEDepict::OESuperAtomStyle::NC, 536
OEDepict::OESuperAtomStyle::NCO, 537
OEDepict::OESuperAtomStyle::Nitrogen, 544
OEDepict::OESuperAtomStyle::NO2, 538
OEDepict::OESuperAtomStyle::OBu, 544
OEDepict::OESuperAtomStyle::OEt, 543
OEDepict::OESuperAtomStyle::Off, 535
OEDepict::OESuperAtomStyle::OMe, 543
OEDepict::OESuperAtomStyle::ONO, 538
OEDepict::OESuperAtomStyle::OPr, 543

OEDepict::OESuperAtomStyle::Oxygen, 544
OEDepict::OESuperAtomStyle::OxygenAndNitrogen, 545
OEDepict::OESuperAtomStyle::Phosphorus, 545
OEDepict::OESuperAtomStyle::PO3H2, 539
OEDepict::OESuperAtomStyle::Pr, 541
OEDepict::OESuperAtomStyle::sBu, 542
OEDepict::OESuperAtomStyle::SO3_, 539
OEDepict::OESuperAtomStyle::SO3H, 539
OEDepict::OESuperAtomStyle::Sulfur, 545
OEDepict::OESuperAtomStyle::tBu, 542
OEDepict::OESuppressNonDepictionHydrogens, 619
OEDepict::OESVGClass, 413
    Constructors, 413
    GetName, 413
OEDepict::OESVGGroup, 413
    AddSVGClass, 414
    Constructors, 414
    GetId, 414
    IsVisible, 414
OEDepict::OETitleLocation, 546
OEDepict::OETitleLocation::Bottom, 546
OEDepict::OETitleLocation::Default, 546
OEDepict::OETitleLocation::Hidden, 546
OEDepict::OETitleLocation::Top, 547
OEDepict::OEWrite2DRingDictionaryReport, 620
OEDepict::OEWriteImage, 621
OEDepict::OEWriteImageToString, 622
OEDepict::OEWriteMultiPageImage, 623
OEDepict::OEWriteReport, 623
OEDepict::OEWriteReportPageByPage, 625
OEDepict::OEWriteReportToString, 625
operator()
    OEDepict::OEDisplayAtomIdx, 229
    OEDepict::OEDisplayAtomMapIdx, 230
    OEDepict::OEDisplayAtomPropBase, 231
    OEDepict::OEDisplayBondIdx, 231
    OEDepict::OEDisplayBondPropBase, 232
    OEDepict::OEDisplayNoAtomProp, 233
    OEDepict::OEDisplayNoBondProp, 234
operator+=
    OEDepict::OE2DPoint, 208

P

PopGroup
    OEDepict::OEImageBase, 311
    OEDepict::OEImageFileBase, 312
PushGroup
    OEDepict::OEImageBase, 311
    OEDepict::OEImageFileBase, 313

R

Render
    OEDepict::OEImage, 294
ringdict2pdf.java
    Example Code, 107
ringdict2pdf.txt
    Example Code, 108

S

SetAddDepictionHydrogens
    OEDepict::OEAlignmentOptions, 213
    OEDepict::OEPrepareDepictionOptions, 381
SetAlignment
    OEDepict::OEFont, 237
SetAntiAliasing
    OEDepict::OEJavaImageBase, 356
SetAromaticStyle
    OEDepict::OE2DMolDisplayOptions, 159
SetAtomColor
    OEDepict::OE2DMolDisplayOptions, 160
SetAtomColorStyle
    OEDepict::OE2DMolDisplayOptions, 161
SetAtomExternalHighlight
    OEDepict::OEHighlightByColor, 259
SetAtomExternalHighlightRatio
    OEDepict::OEHighlightByStick, 271
SetAtomLabelFont
    OEDepict::OE2DMolDisplayOptions, 162
SetAtomLabelFontScale
    OEDepict::OE2DMolDisplayOptions, 164
SetAtomLabelMonochrome
    OEDepict::OEHighlightByBallAndStick, 246
    OEDepict::OEHighlightByCogwheel, 252
    OEDepict::OEHighlightByLasso, 265
    OEDepict::OEHighlightByStick, 272
    OEDepict::OEHighlightOverlayByBallAndStick, 283
SetAtomPropertyFunctor
    OEDepict::OE2DMolDisplayOptions, 168
SetAtomPropLabelFont
    OEDepict::OE2DMolDisplayOptions, 165
SetAtomPropLabelFontScale
    OEDepict::OE2DMolDisplayOptions, 167
SetAtomStereoStyle
    OEDepict::OE2DMolDisplayOptions, 169
SetAtomSVGMarkupFunctor
    OEDepict::OE2DMolDisplayOptions, 170
SetAtomVisibilityFunctor
    OEDepict::OE2DMolDisplayOptions, 171
SetBackColor
    OEDepict::OEPen, 372
SetBackgroundColor
    OEDepict::OE2DMolDisplayOptions, 172
SetBallRadiusScale
    OEDepict::OEHighlightByBallAndStick, 247
    OEDepict::OEHighlightByCogwheel, 253

OEDepict::OEHighlightOverlayByBallAndStick, 283
SetBaseFontSize
OEDepict::OEImageTableOptions, 343
SetBgnPen
OEDepict::OE2DBondDisplay, 137
SetBondColorStyle
OEDepict::OE2DMolDisplayOptions, 173
SetBondLineAtomLabelGapScale
OEDepict::OE2DMolDisplayOptions, 175
SetBondLineGapScale
OEDepict::OE2DMolDisplayOptions, 174
SetBondPropertyFunctor
OEDepict::OE2DMolDisplayOptions, 179
SetBondPropLabelFont
OEDepict::OE2DMolDisplayOptions, 176
SetBondPropLabelFontScale
OEDepict::OE2DMolDisplayOptions, 178
SetBondStereoStyle
OEDepict::OE2DMolDisplayOptions, 180
SetBondSVGMarkupFunctor
OEDepict::OE2DMolDisplayOptions, 181
SetBondWidthScaling
OEDepict::OE2DMolDisplayOptions, 182
SetBoundingBoxPen
OEDepict::OEHighlightLabel, 278
SetButtonFont
OEDepict::OELegendLayoutOptions, 360
SetButtonHeightScale
OEDepict::OELegendLayoutOptions, 361
SetButtonPen
OEDepict::OELegendLayoutOptions, 361
SetButtonWidthScale
OEDepict::OELegendLayoutOptions, 361
SetCellBorderPen
OEDepict::OEImageTableOptions, 343
SetCellColor
OEDepict::OEImageTableOptions, 344
SetCellFont
OEDepict::OEImageTableOptions, 344
SetCellGap
OEDepict::OEImageGrid, 327
OEDepict::OEReportOptions, 405
SetCharge
OEDepict::OE2DAtomDisplay, 129
SetClearCoords
OEDepict::OEAlignmentOptions, 213
OEDepict::OEPrepareDepictionOptions, 381
SetClosed
OEDepict::OE2DPath, 207
SetColor
OEDepict::OEFont, 238
OEDepict::OEHighlightByBallAndStick, 248
OEDepict::OEHighlightByCogwheel, 254
OEDepict::OEHighlightByColor, 260
OEDepict::OEHighlightByLasso, 267
OEDepict::OEHighlightByStick, 273
SetColors
OEDepict::OEHighlightOverlayBase, 280
SetColumnWidths
OEDepict::OEImageTableOptions, 345
SetConsiderAtomLabelBoundingBox
OEDepict::OEHighlightByLasso, 265
SetCoords
OEDepict::OE2DAtomDisplay, 129
SetDefaultBondPen
OEDepict::OE2DMolDisplayOptions, 184
SetDepictOrientation
OEDepict::OEPrepareDepictionOptions, 381
SetDimensions
OEDepict::OE2DMolDisplayOptions, 185
SetDisplayPos
OEDepict::OE2DBondDisplay, 137
SetDisplayType
OEDepict::OE2DBondDisplay, 137
SetEndPen
OEDepict::OE2DBondDisplay, 138
SetExplicitAtomLabelAngle
OEDepict::OE2DMolDisplayOptions, 186
SetFamily
OEDepict::OEFont, 239
SetFill
OEDepict::OEPen, 373
SetFixedCoords
OEDepict::OEAlignmentOptions, 213
SetFont
OEDepict::OEHighlightLabel, 278
SetFontScale
OEDepict::OEHighlightLabel, 279
SetFooterHeight
OEDepict::OEReportOptions, 406
SetForeColor
OEDepict::OEPen, 373
SetHCount
OEDepict::OE2DAtomDisplay, 129
SetHeader
OEDepict::OEImageTableOptions, 346
SetHeaderColor
OEDepict::OEImageTableOptions, 347
SetHeaderFont
OEDepict::OEImageTableOptions, 348
SetHeaderHeight
OEDepict::OEReportOptions, 407
SetHeight
OEDepict::OE2DMolDisplayOptions, 187
SetHPosition
OEDepict::OE2DAtomDisplay, 129
SetHydrogenStyle

OEDepict::OE2DMolDisplayOptions, 188
SetHyperlink
OEDepict::OEFont, 239
SetInnerContour
OEDepict::OEHighlightByCogwheel, 254
SetLabel
OEDepict::OE2DAtomDisplay, 130
SetLabelFont
OEDepict::OE2DAtomDisplay, 130
SetLassoScale
OEDepict::OEHighlightByLasso, 266
SetLegendBoxPen
OEDepict::OELegendLayoutOptions, 361
SetLineCap
OEDepict::OEPen, 373
SetLineJoin
OEDepict::OEPen, 374
SetLineStipple
OEDepict::OEPen, 375
SetLineWidth
OEDepict::OEPen, 375
SetLineWidthScale
OEDepict::OEHighlightByCogwheel, 255
OEDepict::OEHighlightByColor, 260
OEDepict::OEHighlightByLasso, 267
SetMargin
OEDepict::OE2DMolDisplayOptions, 189
OEDepict::OEImageGrid, 327
OEDepict::OEImageTableOptions, 349
OEDepict::OELegendLayoutOptions, 361
SetMargins
OEDepict::OE2DMolDisplayOptions, 190
OEDepict::OEImageGrid, 327
OEDepict::OEImageTableOptions, 350
OEDepict::OELegendLayoutOptions, 362
SetMass
OEDepict::OE2DAtomDisplay, 130
SetMaxAllowedAtomPairClashes
OEDepict::OEAlignmentOptions, 214
SetMaxBondRotations
OEDepict::OEAlignmentOptions, 214
SetMaxPatternMatches
OEDepict::OEAlignmentOptions, 214
SetMinFontSize
OEDepict::OEImageBase, 309
SetOptimizeMacrocycles
OEDepict::OEPrepareDepictionOptions, 382
SetPageHeight
OEDepict::OEReportOptions, 408
SetPageMargin
OEDepict::OEReportOptions, 409
SetPageMargins
OEDepict::OEReportOptions, 409
SetPageOrientation
OEDepict::OEReportOptions, 410
SetPageSize
OEDepict::OEReportOptions, 411
SetPageWidth
OEDepict::OEReportOptions, 412
SetPen
OEDepict::OEHighlightByLasso, 268
SetPerceiveBondStereo
OEDepict::OEPrepareDepictionOptions, 382
SetPosition
OEDepict::OEHighlightByLasso, 269
SetProperty
OEDepict::OE2DAtomDisplay, 131
OEDepict::OE2DBondDisplay, 138
SetPropertyFont
OEDepict::OE2DAtomDisplay, 131
OEDepict::OE2DBondDisplay, 138
SetPropertyOffset
OEDepict::OE2DAtomDisplay, 131
OEDepict::OE2DBondDisplay, 139
SetProtectiveGroupLabelFont
OEDepict::OE2DMolDisplayOptions, 191
SetProtectiveGroupStyle
OEDepict::OE2DMolDisplayOptions, 192
SetRadical
OEDepict::OE2DAtomDisplay, 132
SetRMSDCutoff
OEDepict::OEAlignmentOptions, 215
SetRotateAroundBonds
OEDepict::OEAlignmentOptions, 215
SetRotationAngle
OEDepict::OEFont, 241
SetRowHeights
OEDepict::OEImageTableOptions, 351
SetScale
OEDepict::OE2DMolDisplayOptions, 193
SetSize
OEDepict::OEFont, 242
SetStickWidthScale
OEDepict::OEHighlightByBallAndStick, 248
OEDepict::OEHighlightByCogwheel, 256
OEDepict::OEHighlightByStick, 274
OEDepict::OEHighlightOverlayByBallAndStick, 284
SetStippleFactor
OEDepict::OEPen, 376
SetStubColumn
OEDepict::OEImageTableOptions, 352
SetStubColumnColor
OEDepict::OEImageTableOptions, 353
SetStubColumnFont
OEDepict::OEImageTableOptions, 354
SetStyle
OEDepict::OEFont, 243

SetSuperAtomLabelFont
    OEDepict::OE2DMolDisplayOptions, 194
SetSuperAtomStyle
    OEDepict::OE2DMolDisplayOptions, 195
SetSuppressHydrogens
    OEDepict::OEAlignmentOptions, 216
    OEDepict::OEPrepareDepictionOptions, 382
SetTitleFont
    OEDepict::OE2DMolDisplayOptions, 195
SetTitleFontScale
    OEDepict::OE2DMolDisplayOptions, 197
SetTitleHeight
    OEDepict::OE2DMolDisplayOptions, 200
SetTitleLocation
    OEDepict::OE2DMolDisplayOptions, 199
SetVisible
    OEDepict::OEAtomDisplayBase, 219
    OEDepict::OEBondDisplayBase, 225
SetWidth
    OEDepict::OE2DMolDisplayOptions, 201
SetX
    OEDepict::OE2DPoint, 209
SetY
    OEDepict::OE2DPoint, 209
smartsalign2d.java
    Example Code, 110
smartsalign2d.txt
    Example Code, 111

V

viewmdlsearch.java
    Example Code, 114
viewmdlsearch.txt
    Example Code, 117

W

Write
    OEDepict::OEImageFileBase, 313