



**OpenEye**  
Scientific Software

## **Toolkit Release Notes – Java**

*Release 2012.Feb.1*

**OpenEye Scientific Software, Inc.**

January 11, 2012



# CONTENTS

<b>1</b>	<b>OEToolkits 2012.Feb</b>	<b>1</b>
<b>2</b>	<b>OEChem 1.7.7</b>	<b>3</b>
2.1	Minor bug fixes . . . . .	3
<b>3</b>	<b>OEBio 1.7.7</b>	<b>5</b>
3.1	New features . . . . .	5
3.2	Bug fixes . . . . .	5
<b>4</b>	<b>OESystem 1.7.7</b>	<b>7</b>
4.1	New features . . . . .	7
4.2	Bug fixes . . . . .	7
<b>5</b>	<b>OEPlatform 1.7.7</b>	<b>9</b>
5.1	New features . . . . .	9
5.2	Bug fixes . . . . .	9
<b>6</b>	<b>OEDepict TK 2.0.2</b>	<b>11</b>
6.1	New features . . . . .	11
6.2	Major bug fixes . . . . .	11
6.3	Minor bug fixes . . . . .	12
6.4	Deprecated Ogham . . . . .	12
6.5	Documentation changes . . . . .	12
<b>7</b>	<b>DockingTK 1.1.1</b>	<b>13</b>
7.1	Bug fixes . . . . .	13
<b>8</b>	<b>GraphemeTK 1.0.2</b>	<b>15</b>
8.1	Minor bug fix . . . . .	15
8.2	Documentation changes . . . . .	15
<b>9</b>	<b>GraphSimTK 2.0.1</b>	<b>17</b>
9.1	Minor bug fixes . . . . .	17
<b>10</b>	<b>Lexichem TK 2.1.1</b>	<b>19</b>
10.1	Bug fixes . . . . .	19
<b>11</b>	<b>OmegaTK 2.4.5</b>	<b>21</b>
11.1	New features . . . . .	21

<b>12 SzybkiTK 1.7.0</b>	<b>23</b>
12.1 New features . . . . .	23
12.2 Bug fixes . . . . .	24

# OETOOLKITS 2012.FEB

This is a new release of the OpenEye Toolkits with versions of the following libraries:

**OEChem TK** 1.7.7  
**OEDepict TK** 2.0.2  
**OEDocking TK** 1.1.1  
**Grapheme TK** 1.0.2  
**GraphSim TK** 2.0.1  
**Grid TK** 1.3.5  
**Lexichem TK** 2.1.1  
**MolProp TK** 2.1.2  
**Omega TK** 2.4.5  
**Quacpac TK** 1.5.0  
**Shape TK** 1.8.1  
**Spicoli TK** 1.1.1  
**Szybki TK** 1.7.0  
**Zap TK** 2.1.2

All the toolkits now use an extension of the MMFF94 force field for tricoordinate boron compounds. Please refer to the Omega TK and Szybki TK release notes for a more complete description of what types of boron are supported.



# OEChem 1.7.7

## 2.1 Minor bug fixes

- The `OELibraryGen` class checks the connectivity of each generated product. If any of the atom pairs of a product is connected by more than one bond, for example in molecule `C1C`, then that product is rejected and not returned by the `OELibraryGen.GetProducts` method. [FogBugz #9759]
- Fixed a bug that would invalidate an atom iterator after deleting the atom the iterator currently points at. This would only occur on molecules that already had deleted atoms when the iterator was constructed from a `GetAtoms` method that takes a predicate.
- `oemolstreambase.GetFlavor` will no longer crash when given a format larger than `OEFormat.MAXFORMAT`.
- `OEAddMols` will now set the dimension of the left hand side to the dimension of the right hand side if the dimension of the left hand side is zero. This allows an empty molecule to be used to accumulate 3 dimensional molecule without losing the dimension.
- Fixed a bug in the procedure for generating atom names for PDB files that could cause a crash when some atoms have very high atom indices.



# OE BIO 1.7.7

## 3.1 New features

- Added the `OESymmetryFractionalToCartesian` and `OESymmetryCartesianToFractional` functions to swap between crystal coordinates and cartesian coordinates.
- Added the `OEBuildUnitCell` function to apply the symmetry operations to the asymmetric unit cell to generate the complete unit cell given a molecule and a spacegroup.
- `OESetCrystalSymmetry` now has a force flag that sets the symmetry parameters even if the cell restrictions are violated.

## 3.2 Bug fixes

- `OECopyCrystalSymmetry` now follows standard OpenEye guidelines for argument position and naming to always have the output parameter first. This is an API breaking change, code that previously used `OECopyCrystalSymmetry` will no longer work.
- Slightly reduced memory footprint of symmetry operations.
- Crystal spacegroup names from PDB files are now handled more robustly when searching for CCP4 long and short names.
- Added symmetry operations for the P21/n space group. [FogBugz #11796]



# OESYSTEM 1.7.7

## 4.1 New features

- Adding the `OESystem.OETransparentColor` color constant that is used in *OEDepict* to generate PNG images with a transparent background.
- The `OEGeom*` functions should now be fully wrapped in the target languages: Python, Java, and C#.

## 4.2 Bug fixes

- The `OEPreIter` increment and decrement methods will now always move the underlying iterator before checking the validity of the underlying iterator. This fixed the `OEChem` bug that invalidated an atom iterator after the current atom was deleted.
- Fixed a memory leak inside `OEIter.Push` that takes a predicate.
- `OECopyBaseData` used to call `Clear` on the destination object before copying the data. This could clear an entire molecule instead of the just the generic data. `OECopyBaseData` will now only delete the existing generic data.



# OEPLATFORM 1.7.7

## 5.1 New features

- `OEIsNormalNum` added to return whether a floating point number is normalized, denormalized, or zero.
- `OEGetProcessMemUsage` added to return the amount of memory being used by the current process.

## 5.2 Bug fixes

- `OEIsNaN` fixed to actually work when the compiler was passed `-ffast-math`. Also, it will now return `true` whenever passed `NaN` or `Infinity`.
- `OEAllocate` and `OEDeallocate` were deprecated long ago and are now removed.



# OEDEPICT TK 2.0.2

## 6.1 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`) [FogBugz #11868 and FogBugz #12712]
- Adding a scaling factor that can be used to increase or decrease the size of the fonts used to depict bond properties. (`OE2DMolDisplayOptions.SetBondPropLabelFontScale`) [FogBugz #11868]
- 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. [FogBugz #12040 and FogBugz #12712]
- Fixing the run-time initialization of the color that is used to depict super atoms. [FogBugz #12873]

## 6.2 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)

## 6.3 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. [FogBugz #12485]
- `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. [FogBugz: #12411]
- 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 `OEAtomicStyle.Circle` style, a check is performed to insure that the detected ring is not a fused ring of two smaller rings. [FogBugz #11858]
- 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 `OEPrepAlignedDepiction` function, the fit molecule is scaled by the average bond length of the reference molecule. [FogBugz #12465]

## 6.4 Deprecated Ogham

- Deprecating the `OEDepictFixedCoordinates` function. For molecule alignment, please use the `OEPrepAlignedDepiction` function.

## 6.5 Documentation changes

- Renaming the Application chapter to *OEDepict Examples Summary* and inserting the source code of the examples into the *chapter\_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.

# DOCKINGTK 1.1.1

## 7.1 Bug fixes

- Fixed bug where SD data on the input molecules was not copied to the docked output molecule.
- Fixed memory leak when scoring a molecule that does not fit within the receptor site.
- Fixed an invalid memory access bug that occurred in rare cases when docking molecules.



# GRAPHEMETK 1.0.2

## 8.1 Minor bug fix

- Avoiding extraneous runtime warnings from `OEDrawPearlsCircle` and `OEDrawFlowerCircle` when the pen's fill is turned off.

## 8.2 Documentation changes

- Some examples and images were updated in order to consistently use red and blue color for representing negative and positive values respectively when using the `OELinearColorGradient` 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.



# GRAPHSIMTK 2.0.1

## 9.1 Minor bug fixes

- If a molecule being fingerprinted has only hydrogen atom(s), for example molecule `[H]`, then an empty fingerprint is generated, that is, all bits of the fingerprint are set to zero. All hydrogens of the molecule (polar, stereo, isotope, charge) are suppressed before generating its fingerprint.
- If the shell of the circular fingerprint can not be extended, for example, all atoms of the molecule are already being considered, then the search is terminated. This modification does not effect the generated circular fingerprints, it only makes the generation process faster.
- Some of the algorithms generating fingerprints can return the same fragment of a molecule more than once. For example the path fingerprint enumerates both `OC` and `CO`. These duplicates are now filtered out when calling the `OEGetFPOverlap` function. Two overlaps are considered equivalent only if they store the same target and pattern atoms and bonds, not considering the correspondence between the pattern and target atoms and bonds.



# LEXICHEM TK 2.1.1

## 10.1 Bug fixes

- Fixed a bug with structure generation for tetramethylammonium pyrrolidine-2-carboxylate trihydrate.
- Components of salts are now separated by semi-colons.
- Added numerous dictionary entries.
- Added additional ring templates.
- Fixed a bug with names including benzene) or benzene}, the appropriate substitution should have been phenyl.
- Fixed an issue when incorrect von Baeyer names are parsed to Lexichem, Lexichem now fails silently.
- Fixed dictionary lookup algorithm.
- Fixed a crash in Mol2Nam when a H-atom was part of the main ring system in simple spiro compounds.
- Fixed a bug in names like sulfuric acid mono(x) ester.



## OMEGATK 2.4.5

### 11.1 New features

- Now uses an extension of the MMFF94 force field for tricoordinate boron compounds. Most compounds containing B-X bonds where X=C,N,O,S, and H are covered with the following exceptions: X=N(imine),N(sulfonamide), N(pyridinium) and N(quaternary). Also not supported are compounds in which boron is bonded to X=F,Cl,Br,I,B and Si, or makes a bond angle BYX. Compounds in which boron is a part of four-membered rings of B1CCC1 type are also not available in the current parameterization because their existence is questionable: Ab initio calculations at the MP2/6-31G\*\* level failed to identify stable structures for them (highly polar structures in which boron is four-coordinated are formed).



# SZYBKITK 1.7.0

## 12.1 New features

- An extension of the MMFF94 force field for tricoordinate boron compounds is offered in this release. Most compounds containing B-X bonds where X=C,N,O,S, and H are covered with the following exceptions: X=N(imine),N(sulfonamide), N(pyridinium) and N(quaternary). Also not supported are compounds in which boron is bonded to X=F,Cl,Br,I,B and Si, or makes a bond angle BYX. Compounds in which boron is a part of four-membered rings of B1CCC1 type are also not available in the current parameterization because their existence is questionable: Ab initio calculations at the MP2/6-31G\*\* level failed to identify stable structures for them (highly polar structures in which boron is four-coordinated are formed).
- NOTE WELL: Because of the partial parameterization for boron-containing compounds, users of the OEmmff class (derived from OEGenericFF) need to pay attention to the return value of the OEGenericFF.Setup method: In the case where the parameters for a specific boron molecule are not available, this function returns false. Checking the return value of OEGenericFF.PrepMol is not enough, because it does not catch the case of missing force field parameters for a specific compound.
- Two new API methods have been added which control the salt concentration for all PB calculations with SzybkiTK. These are: OESzybki.SetSaltConcentration and OESzybki.GetSaltConcentration. The first method takes salt concentration in M as a float number. The valid range is 0 - 0.08M; this method should not be used for higher salt concentrations. The second method returns the current salt concentration as a float. Default value is zero. Previous versions of SzybkiTK assumed zero salt concentration.
- All PB calculations carried out with the previous SzybkiTK versions used Bondi atomic radii. The current release offers two additional sets of atomic radii, called ZAP7 and ZAP9 as alternatives. They are described by Nicholls et. al. The new method: OESzybki.SetAtomicRadii allows the use of one of these two sets. The new method OESzybki.GetAtomicRadiiType returns the type of current atomic radii set.
- Better control of dielectric constants is provided. This includes a new method: OESzybki.SetSolventDielectric which allows change from the default value of 80. Method OESzybki.GetSolventDielectric returns the current value of the solvent dielectric constant. Method: OESzybki.SetSolventModel which determines the solvation model for a free ligand in solution, now takes an additional default parameter which sets the intrinsic dielectric constant of the ligand.
- A method which obtains gradients from the single point calculations is added. Specifically, a new method OESzybkiResults.GetGradients is added to the szybki\_results.h header, and two new methods to the szybki.h header file. There are: OESzybki.SetCalculateGradients and OESzybki.GetCalculateGradients.
- Enforcement of proper behavior by the OESzybki.operator for molecules without 3D coordinates. Such molecules are not processed and a warning is issued.

## 12.2 Bug fixes

- Method `OESzybki.GetProteinBoundLigandEntropy` is protected from selection of an inappropriate protein-ligand electrostatic model (entropy calculations for protein-bound ligands are done with the `OEProteinElectrostatics.ExactCoulomb` model).
- Calling `OESzybki.SetRunType` with an argument `OERunType.CartesiansOpt` immediately after a call to same method but with an argument `OERunType.TorsionsOpt` caused missing potential values for bonds and bond angles. The problem is fixed.
- Method `OESzybkiResults.Print` was not reporting MMFF terms in the case when `OESzybki.SetProtein` was used. The bug is fixed. Also, some minor improvements to this method is made, so it is clear which terms contribute to the total protein-ligand interactions and which are reported only for comparison. For example the usage of `OEProteinElectrostatics` results in reporting the exact and approximated Coulomb terms, however only the latter is used in the reported protein-ligand interaction.
- Fixed a minor bug where unnecessary warnings regarding missing protein parameters were issued in an entropy calculation where the protein is held rigid. This happens when mistakes or inaccuracies in the input protein structures lead to undefined force field parameters.
- Previously, processed molecules were output with aromaticity specified according to the MMFF94 aromaticity model. Now, output molecule are converted to the OE aromaticity model.
- Entropy calculations using the quasi-Newton method now give significantly more accurate answers in the two extreme cases where the input ligand structure is a) already optimized, or b) has a very high-energy (i.e. poor) structure.