



Flare™ V3.0 Release Notes

About Flare	2
Supplied Binaries	4
Supported Platforms	4
Minimum specifications	4
Recommended specifications.....	4
Supported Operating Systems.....	5
GPU Calculations	5
Installation.....	5
Windows.....	5
Linux	5
macOS.....	5
Start Flare	6
Windows.....	6
Linux	6
macOS.....	6
Licensing	6
Changes in version 3.0.....	6
Changes in the Flare GUI	6
Changelog for the Flare Python API	11
Changes in version 2.0.1.....	14
Changes in version 2.0.....	14
Version 1.0.....	15
Uninstalling Flare.....	16
Windows.....	16
Linux	16
macOS.....	16
Known Issues and Bugs.....	16
Third party software.....	17
Support	17

About Flare

[Flare](#) is a structure-based design tool for the visualization and analysis of protein-ligand complexes.

Capabilities in Flare V3.0 include:

- Smart import and handling of protein-ligand complexes
- Accurate and reliable protein preparation
- Sequence alignment and superimposition
- Molecular Dynamics
- Minimization of protein ligand complexes
- Accurate docking using Lead Finder™
- Ligand-based conformation hunt and alignment
- Free Energy Perturbation simulations
- Protein interaction potentials
- Electrostatic Complementarity™
- 3D-RISM water analysis
- WaterSwap analysis for ligand and binding site energetics
- The Flare Python™ API

Docking and scoring can be performed in Flare using BioMolTech's Lead Finder docking algorithm,¹ now fully integrated in the GUI. The docking engine in Lead Finder combines a genetic algorithm search with local optimization procedures, which make Lead Finder efficient in coarse sampling of ligand poses and following refinement of promising solutions. Three different scoring functions are employed and optimized for the accurate prediction of 3D docked ligand poses, protein-ligand binding energy and rank-ordering of active and inactive compounds in virtual screening experiments.

Ensemble docking is a module of Flare for considering active site flexibility when alternative active site conformations of the same protein are available. Docking is performed within a workflow which enables the inclusion of multiple proteins in the same docking run, each corresponding to a different conformation of the active site.

In both traditional and ensemble docking experiments, template docking is useful when you know the pose of a 'template' ligand, and wish to use this information to bias the docking results for congeneric compounds. The molecules to be docked are aligned by substructure to the template ligand, and the aligned conformation is used to seed the docking run, generally leading to improved docking results.

Covalent docking is a module of Flare for predicting the binding pose and interactions of covalent inhibitors, a class of ligands which derive their activity by forming a covalent bond with the target while making at the same time a network of non-covalent interactions with the active site.

Ligand-based alignments can be used in Flare as a way to make meaningful comparisons across chemical series, and to generate sensible poses within a protein active site. Ligands can be aligned by using the ligand fields, by using shape properties or by using a common substructure. Using the fields gives a 'protein's view' of how the molecules would line up in the active site, generating ideas on how molecules with different structures could interact with the same protein. Using substructure or common shape properties shows how the fields around a single chemical series vary with activity.

Molecular Dynamics using OpenMM² can be used to study the conformational changes of proteins and the stability of ligand-protein complexes.

Free Energy Perturbation (FEP) calculations allow for the prediction of relative binding affinity changes within a congeneric ligand series. In these calculations a molecule is gradually converted into a structurally closely related analogue via a non-physical ('alchemical') pathway. By assessing the free energy difference ($\Delta\Delta G$) between the end states of such transformations, accuracies of about 1 kcal/mol compared to experimental values can be achieved for large datasets. FEP in Flare is developed in collaboration with Julien Michel *et al.* at the University of Edinburgh.

Flare calculates and displays protein interaction potentials (protein fields) based on Cresset's XED force field.^{3,4} This methodology is extremely useful to understand the electrostatic environment of the active site of a protein, comparing protein and ligand electrostatics to improve new molecule design. Protein interaction potentials can also be used to analyze the active site of closely related proteins, to understand how subtle changes in the 3D conformation of the proteins, or in the protein sequence, can influence the electrostatic patterns of the active site. The analysis of the electrostatic patterns of a protein family using protein fields is also useful to inform the design of more selective ligands.

The Electrostatic Complementarity⁵ coloring and scoring functions within Flare are an extension of the protein interaction potential functions. They provide a detailed map and a quantitative score of how well the electrostatics of one or more ligand(s) matches that of the protein active site and can accordingly prove invaluable for the understanding of ligand binding, SAR and the design of new molecules that target the protein.

3D-RISM water analysis,⁶ based on the XED force field, investigates the location and stability of water molecules in your protein. This information can be used to identify energetically unfavorable ('unhappy') molecules which should be possible to displace from the active site by designing a ligand with the correct shape and electrostatic features. Incorporating energetically favorable ('happy') water molecules in the analysis of the active site using protein interaction potentials will lead to a better understanding of the electrostatic requirements of ligand-protein interaction, and to improved ligand design.

WaterSwap^{7,8} is a Monte Carlo technique for investigating ligand-protein interactions, which allows the calculation of absolute protein-ligand binding free energies from first-principles, condensed-phase simulations. This method, developed by Chris Wood *et al.* at the University of Bristol, is fully integrated into Flare. Flare also supports advanced processing solutions for remote calculation.

The Flare Python API enables Flare functionality to be accessed from Python and for you to customize the Flare interface. Python scripts can be run from the Flare graphical user interface (GUI) or by the command line `pyflare`.

1. O. V. Stroganov *et al.*, Lead Finder: an approach to improve accuracy of protein-ligand docking, binding energy estimation, and virtual screening, *J. Chem. Inf. Model.* **2008**, *48*, 2371-2385
2. <http://openmm.org/>
3. J. G. Vinter, Extended Electron Distributions Applied to the Molecular Mechanics of some Intermolecular Interactions, *J. Comput.-Aided Mol. Des.* **1994**, *8*, 653-668
4. J. G. Vinter, Extended electron distributions applied to the molecular mechanics of intermolecular interactions II - Organic Complexes, *J. Comput.-Aided Mol. Des.* **1996**, *10*, 417-426
5. M. R. Bauer and M. D. Mackey, Electrostatic Complementarity as a Fast and Effective Tool to Optimize Binding and Selectivity of Protein-Ligand Complexes, *J. Med. Chem.* **2019**, *62*, 3036 -3050
6. R. Skyner *et al.*, A Review of Methods for the Calculation of Solution Free Energies and the Modelling of Systems in Solution, *Phys. Chem. Chem. Phys.* **2015**, *17*, 6174-6191

7. C. J. Woods *et al.*, A water-swap reaction coordinate for the calculation of absolute protein-ligand binding free energies, *J. Chem. Phys.* **2011**, *134*, 02B611,
8. C. J. Woods *et al.*, Rapid Decomposition and Visualisation of Protein-Ligand Binding Free Energies by Residue and by Water, *Farad. Discuss.* **2014**, *169*, 477-499

Supplied Binaries

Flare functionality is delivered through a graphical user interface (GUI) with two interlinked binaries (Flare, FieldEngine). However, the GUI will call other binaries to perform specific tasks.

Flare	The main graphical interface for Flare
pyflare	Python scripts can be run from Flare GUI or by the pyflare command line tool. This binary gives you access to all Flare methods directly from the command line enabling you to automate and script your workflows.
FieldEngine	<p>The calculation engine for Flare. FieldEngine is started directly by the primary interface to perform specific calculation tasks. By default, one FieldEngine binary is started for each CPU core on the host machine (but this can be reconfigured in the application preferences). The applications communicate with the FieldEngine using a TCP/IP protocol that can cause network access warnings in some situations.</p> <p>The FieldEngine process can be run on a remote machine with communication managed through a variety of methods including Cresset Engine Broker™.</p>

Supported Platforms

Minimum specifications

Hardware	Specifications
RAM	4GB
Processor	Intel i5 processor or equivalent
Disk space	1GB

Recommended specifications

Hardware	Specifications
RAM	16GB
Processor	Intel i7 processor or equivalent
Disk space	1GB

Supported Operating Systems

Flare is supplied as 64bit binaries for Windows®, macOS® and Linux®.

Windows®	macOS®	Linux®
Windows 10	macOS 10.14	Red Hat Enterprise Linux 7

In addition, running FEP calculations is only supported on Linux.

GPU Calculations

Flare's Dynamics and FEP calculations require GPUs which support OpenCL or NVIDIA CUDA®. NVIDIA CUDA requires additional software to be installed, please refer to [NVIDIA's CUDA installation instructions](#) for more details.

Installation

Platform-dependent installation instructions are given below and at <https://www.cresset-group.com/support/support-resources/installing-desktop-applications>.

Windows

1. Browse to the Windows folder of the Flare distribution media (or locate the installer in your Downloads folder) and double click the executable installer file.
2. Follow the on-screen instructions.
3. If you have a previous version of Flare installed, then this will be removed before installing the new version.

Linux

1. Browse to the Linux folder of the Flare distribution media.
2. Install the Flare re-locatable rpm package using one of the command lines below (where <vers> is the version number, <arch> is the architecture and <path> is install location (/opt by default):

```
yum install cresset-flare-<vers>-1.<arch>.rpm
```

or

```
rpm -U [--prefix <path>] cresset-flare-<vers>-1.<arch>.rpm
```

3. Alternatively, unpack the supplied tgz file to a suitable location (e.g. \$HOME or /usr/local or /opt).

macOS

1. Open the supplied disc image.
2. Drag the Flare icon onto the Applications icon to install the application in the Applications folder.

Start Flare

Windows

The install process places an entry in the 'Start menu' and a shortcut on your desktop.

Double click the Flare shortcut to start the application.

Alternatively, in the search box on the taskbar type Flare (the search panel opens automatically as you type). In the search results, click Flare to start the application.

Linux

If the installation was from rpm then select the shortcut in the Applications menu.

Alternatively, in a terminal window type:

```
/path_to_installation/cresset/Flare/bin/Flare
```

where *path_to_installation* is the directory where the Flare distribution was unpacked. If Flare was installed from an rpm package keeping the default location, then the installation will be in `/opt`.

macOS

Navigate to the Applications folder and double click the Flare icon.

Licensing

To run Flare, you will require a valid license file which will be supplied by Cresset, usually as an attachment to an email. If you cannot locate your license file, then please contact support@cresset-group.com.

To install a license, start the application and click on the 'Install license' button. Alternatively, the license file can be manually placed in one of the directories detailed at <https://www.cresset-group.com/support/support-resources/license-file-locations/>.

Changes in version 3.0

Changes in the Flare GUI

- New Free Energy Perturbation ('FEP') module in the Ligand tab, to predict relative binding affinity changes within a congeneric ligand series.
- New 'Covalent Docking' module in the Ligand tab, for predicting the binding pose and interactions of covalent inhibitors
- New template docking option available for standard and ensemble docking experiments, useful in those cases where the pose of a 'template' ligand is known and can be used to bias the docking results for congeneric compounds
- New 'Dynamics' module in the Protein tab, using OpenMM to study the conformational changes of proteins and the stability of ligand-protein complexes
- New 'Conf Hunt and Align' module in the Ligand tab, to perform ligand-based alignment
- GAFF2 force field now available for 3D-RISM, WaterSwap, Dynamics and FEP

- New Tiles view showing a compacted version of the Ligands table
- New Sequence Similarity table reporting the results of sequence alignment in a matrix format
- New and improved editing of ligands, proteins and waters in the 3D window, including all the Molecular Editor functionality of previous Flare releases, enabling at the same time full access to ligand and protein selection and appearance changes functionality. Also including:
 - Full access to all the periodic table elements
 - New button to edit protein residue information
 - New button to fix and reorder protein residues based on current connectivity
 - New option to edit a selected portion of a molecule
 - Relative manual movements of proteins now possible
- New functions and options for the Home tab:
 - Ribbon colors menu, to color protein ribbons independently from the color of carbon atoms, and to color ribbons by residue, protein sequence, protein chain and protein temperature factor
 - Option in the 'Ribbon' button to set ribbon transparency
 - Ribbons can now be displayed also for DNA and RNA
 - Atom colors option to color also non-carbon atoms with a solid color
 - Option in the Atom Pick panel to pick atoms by name
 - Option in the Atom Pick panel to find specific residue sequences
- New functions and options for the Ligand tab:
 - 'New' button to create new ligands from scratch
 - 'Constraints' button for adding or removing docking constraints to picked protein atoms, or to set ligand constraints for alignment and virtual screening
 - 'Send to Blaze' button to run new Blaze searches from Flare
 - 'Download Blaze Search Results' button to retrieve the results of existing Blaze searches
 - 'Remove all Tags' option to remove all tags from selected molecules
 - 'Diff' button to show field differences for two selected ligands
 - New customizable option in the standard, ensemble, template and covalent docking calculation panels to set the number of independent docking runs for each ligand, keeping the best poses overall
- New 'Show Secondary' button in the Proteins tab to show a colored line above the residue names in the Proteins and Alignment tables, corresponding to the secondary structure
- New functions and options for the View tab:
 - 'Synchronize Selections' button to keep in sync the selection of ligands and associated proteins
 - Option to choose the screen to use for Full Screen mode

- Option to set the stereo method for the current project
- New Help tab to access Flare documentation and information, giving access to:
 - Flare manual and Flare Python Developers Guide
 - 'Tips and Tricks' page on the Cresset website
 - Information about the Flare version and relevant licensing information, including the Flare license expiry date
 - Cresset website
 - Flare release notes
 - Online form to register to the Cresset newsletter
 - email to Cresset support
 - Generation of diagnostic data
- New filter button in the Alignment table to show only protein chains belonging to the desired alignment group
- New right-click options in the Proteins and Alignment tables, including:
 - Choose a specific protein chain to set the residue sequence numbering in the ruler
 - Select a protein's associated ligands
 - Invert the current selection of proteins
- New right-click options for the Ligands table, including:
 - Set ligand constraints for alignment and virtual screening
 - Select a ligand's associated protein
 - Clear the protein association
 - Invert the current selection of ligands
 - Export selected docked poses and alignments
 - Convert conformation to poses
 - Choose any user column as the molecule title
 - Choose the column to use as a label in the 3D window
 - Set user column type (text/number)
- New advanced options to disable small side chain movements and to remove atoms from residues with incomplete backbone during protein preparation
- New right-click options for the 3D window:
 - Add custom labels/annotations to ligand and protein atoms
 - Select ligands and proteins
 - Add or remove docking constraints to picked protein atoms

- New import and export of Flare Preferences from the command line
- New Appearance Preferences options:
 - Customizable option to limit the maximum number of ligands to display, giving a warning when the maximum number of ligands is exceeded
 - Option to disable pink halo highlighting atoms under the mouse cursor
 - Option to change the default green=good and red=bad colors for good/poor electrostatic complementarity, favorable/unfavorable 3D-RISM waters, favorable/unfavorable interaction free energies for WaterSwap into a custom color
 - Maximum copy picture size resolution can now be set to 10x
- New Calculations Preferences panel, to set the maximum number of alignments to generate for each ligand
- New Table Preferences panel, to customize visible/hidden columns in the Ligand table and visible/hidden tile items in the Tiles view
- New option to use all available local CPUs, relaxing the 16-CPU limitation in previous Flare releases
- New Blaze Preferences panel, to set-up the communication with Blaze
- Improved and faster XED minimization algorithm
- Improved handling of metals in the XED force field – most common metals (Zn, Fe, Mg, Ca etc.) found in a protein environment should be modelled more accurately
- Improved atom typing during protein preparation
- Improved handling of DNA and RNA
- Improved handling of larger datasets
- Improved Electrostatic Complementarity algorithm, including:
 - Restricting the EC calculation to regions of the ligand surface which are in contact with the protein
 - Calculation of Electrostatic Complementarity scores for ligand alignments
 - Electrostatic Complementarity scores are now calculated against each associated protein for ensemble docking poses
- Improved ‘Accurate but Slow’ and ‘Very Accurate but Slow’ docking calculation methods for standard, ensemble, template and covalent docking, now performing 3 independent docking runs and keeping the best poses overall
- Improved ‘Score Only’ calculation method mode for standard, ensemble, template and covalent docking, now giving the option to disable pose optimization
- Improved handling of conjugated bonds in Lead Finder
- Improved docking processing panel, now including by default the ion and cofactors chains in the docking experiment

- Updated AmberTool version to AmberTools 2019
- Updated Lead Finder and Build Model algorithms
- Updated Sire to version 2019.1.0
- Updated RDKit to version 2019.03.4
- Improved Contacts panel, including:
 - New contact definitions for weak H-bonds and salt-bridge interactions
 - Full control on the interactions to show, with fully customizable matrixes for favorable and unfavorable interactions
 - New option for keeping the matrixes for favorable and unfavorable interactions in sync
 - New option to show the interactions in bold style
 - New option to show only interactions between associated molecules
- Improved Filters window, now including:
 - A new green/red toggle to control whether each filter is enabled or disabled
 - A new settings button enabling import and export of the set filters
- Improved filtering options for text columns, now including 'equal', 'not equal', 'containing' and 'not containing' options
- Improved structural filters now including pre-defined filters for Ring, Aromatic Ring, Non-ring atom, Chiral atom, H-bond donor and H-bond acceptor
- Improved Processing Preferences panel, including a completely redesigned GUI for setting and customizing multiple Cresset Engine Brokers, and a section to set local GPU processing
- Improved PDB download widget now supporting multiple PDB download paths
- Improved Radial Plots and Multi-Parameter Scoring, including:
 - New settings button in the Radial Plots Properties window, enabling import and export of the radial plot properties profile, and to restore the radial plot properties to default values
 - New option to specify whether invalid values for each property should be shown at the edge or at the center of the radial plot
- Improved 2D display of molecules
- Improved 'Move to Protein' button in the Ligand tab, now allowing to move a selected ligand to a new protein in the Proteins table
- Proteins can now be sorted and reverse-sorted by protein title in the Proteins and Alignment tables
- Improved way for drawing of sequence alignment constraints
- Calculation and display of protein surfaces is now performed by default on visible atoms
- Improved export of molecules with drag-and-drop, now exporting all Ligands table data as sdf tags.

- Improved 'Atoms' button to show/hide atoms and bonds, now giving the option to show only picked atoms
- Improved 'Color by residue' function, now coloring atoms and ribbons by residue type
- Smoother transition across Storyboard scenes
- Improved layout of all Flare ribbon menus

Changelog for the Flare Python API

New Classes:

cresset.flare.BatchEdit

cresset.flare.Conformation

cresset.flare.ConformationList

cresset.flare.DockingInfo

cresset.flare.DynamicsResult

cresset.flare.FepProject

cresset.flare.FepProjectList

cresset.flare.FepNode

cresset.flare.FepNodeList

cresset.flare.FepLink

cresset.flare.FepLinkList

cresset.flare.FepStructure

cresset.flare.RadialPlotProperty

cresset.flare.RadialPlotPropertyList

New Calculations:

cresset.flare.Align

cresset.flare.ConformationHunt

cresset.flare.Dynamics

cresset.flare.DynamicsOpenMM

cresset.flare.DynamicsAmber

cresset.flare.Fep

cresset.flare.FepOpenMM

cresset.flare.FepAmber

cresset.flare.FepGenerateLinks

New Callbacks:

cresset.flare.CameraChangedCallback
cresset.flare.EditingMoleculeChangedCallback
cresset.flare.LigandPropertyChangedCallback
cresset.flare.PosePropertyChangedCallback

New Methods:

cresset.flare.Camera.restore_view
cresset.flare.Camera.save_view
cresset.flare.LeadFinderSystem.generated_grid
cresset.flare.Ligand.align_to
cresset.flare.Ligand.batch_edit
cresset.flare.Ligand.recharge
cresset.flare.LigandList.sort
cresset.flare.Pose.align_to
cresset.flare.Pose.batch_edit
cresset.flare.Pose.recharge
cresset.flare.PoseList.sort
cresset.flare.Protein.batch_edit
cresset.flare.Protein.recharge
cresset.flare.ProteinList.sort
cresset.flare.RoleList.sort
cresset.flare.Sequence.find_sub_sequence

Improved Methods:

cresset.flare.ProjectLogList.append: added optional extra arg molecules
cresset.flare.ProjectLogList.insert: added optional extra arg molecules
cresset.flare.ProteinSurfaceList.new: added kwargs for quality and threshold

New Properties:

cresset.flare.Project.fep_projects
cresset.flare.WaterSwapAmber.gaff_version
cresset.flare.Rism.gaff_version
cresset.flare.Ligand.conformations
cresset.flare.MainWindow.picked_residues

cresset.flare.ProteinFieldSurface.back_clip
cresset.flare.ProteinFieldSurface.front_clip
cresset.flare.ProteinFieldSurface.sphere_clip
cresset.flare.ProteinMolecularSurface.back_clip
cresset.flare.ProteinMolecularSurface.front_clip
cresset.flare.ProteinMolecularSurface.sphere_clip
cresset.flare.ProteinRismSurface.back_clip
cresset.flare.ProteinRismSurface.front_clip
cresset.flare.ProteinRismSurface.sphere_clip
cresset.flare.Camera.ligand_field_diff_enabled
cresset.flare.AtomStyle.ribbon_color
cresset.flare.MoleculeStyle.ribbon_color
cresset.flare.AtomStyle.annotation_label_visible
cresset.flare.MoleculeStyle.annotation_label_visible
cresset.flare.AtomStyle.annotation
cresset.flare.MoleculeStyle.annotation
cresset.flare.Protein.logs
cresset.flare.Ligand.logs
cresset.flare.ProjectLog.molecules
cresset.flare.ProjectLog.datetime
cresset.flare.Ligand.references
cresset.flare.ProteinPrep.remove_atoms_from_residues_with_incomplete_backbone
cresset.flare.ProteinPrep.allow_small_side_chain_movements
cresset.flare.LeadFinderSystem.template_ligand
cresset.flare.LeadFinderSystem.number_of_runs
cresset.flare.LeadFinderSystem.covalent_residue
cresset.flare.LeadFinderSystem.docking_constraints
cresset.flare.Project.radial_plot_properties

Improved Properties:

cresset.flare.Atom.name is now writeable

Changes in version 2.0.1

- Windows only release, to fix an issue causing an incorrect update of the 3D window. This issue only affects systems with an Intel GPU and drivers version 24.* or later on the Windows operating system.

Changes in version 2.0

- New Flare Python API enabling Flare functionality to be accessed from Python and customization of the Flare interface. Python scripts can be run from the Flare graphical user interface (GUI) or by the command line `pyflare`
- New Python tab in the Flare GUI providing access to the Flare Python API functions within the Flare GUI
- New Electrostatic Complementarity Scores, calculated for each ligand based on the associated protein and saved as columns in the Ligands table. These provide three alternative quantitative scores of how well the electrostatics of one or more ligand(s) matches that of the associated protein active site
- New electrostatic complementarity surfaces for ligand and proteins, providing a detailed visual map of ligand/protein electrostatic complementarity
- New electrostatic potential surfaces for ligands and proteins, colored according to the electrostatic potential calculated with the XED force field
- New coloring of protein surfaces by secondary structure and Wimley-White residue hydrophobicity
- New Ensemble Docking functionality, for considering active site flexibility when alternative active site conformations of the same protein are available
- New Filter window, to show in the Ligands table only the ligands that conform to a desired set of rules. Includes filtering on numerical values, text data, Boolean values, tags, ligand structure using either a SMARTS string or a substructure sketched into the Flare Molecule Editor
- New Storyboard window, to capture scenes recording all details from the 3D window that can easily be recalled when needed, including the capability to annotate and rename scenes
- New Radial Plot in the Ligands table to support Multi-Parameter Scoring
- New 3D-RISM Surfaces button, to recalculate and display 3D-RISM surfaces when a completed 3D-RISM analysis is available for the selected proteins
- New 'Promote Pose to Ligand' menu button to promote a chosen docked pose to ligand status
- New 'Delete Docking Grid on Selected Proteins' function to delete unused docking energy grids, reducing the amount of disk space in the saved Flare project
- New spherical clipping of protein surfaces
- New border text displaying relevant information at the edge of the 3D window, especially useful when working in grid mode
- New 'Export all proteins' functionality
- New export file format for proteins as PDB files with Amber conventions
- New export of ligand field surfaces

- New 'Move to protein' menu button to merge a ligand to a chosen protein structure
- New button in Flare Appearance Preferences to swap color of Cresset fields (blue/red)
- Updated Lead Finder and Build Model algorithms
- Updated sire (which contains WaterSwap) to version 2018.1.0
- Updated AmberTools to version 18
- Updated RDKit to version 2018.03.1
- Improved protein superimposition now allowing the superimposition of all proteins vs. selected proteins
- Improved coloring of crystallographic water molecules according to calculated 3D-RISM ΔG
- Improved WaterSwap workflow
- Improved weighting scheme for consensus calculated free energy of binding from WaterSwap runs
- Improved functionality of Protein Surfaces window, now enabling the update of saved protein field and molecular surfaces in terms of transparency, quality and chains included
- Improved definitions for ligand-protein interactions, as fully documented in the Flare manual
- Improved import of peptides as ligands
- The FieldEngine binary now enables the setting of the maximum number of cores the process should use
- Improved performance of multi-threaded docking runs
- Improved layout of the functions in the Protein tab
- Improved display of protein entries in the Protein table: the protein name is now the PDB code/file name for the protein; name column is resizable; background color for the 'visibility' icon matches the color of protein carbon atoms
- Improved right-click menus
- Improved 'mark as favorite' behavior for ligands
- Improved functionality for moving all/chosen ligand(s) from protein(s) into the Ligands table
- Improved logging for most functions
- Improved ligand minimization, protein preparation dialogs
- Improved clipping of front and back Z-planes in the 3D window, now allowing to choose whether the ligands should be affected by the clip planes
- Improved Flare shortcuts
- Lots of additional improvements and bug fixes.

Version 1.0

- Initial release of Flare.

Uninstalling Flare

Windows

Flare can be uninstalled using the standard 'Programs' Control Panel shortcut.

1. In the search box on the taskbar, type Control Panel and select it from the results.
2. Select Programs > Programs and Features.
3. Press and hold (or right-click) on the Flare icon and select Uninstall, then follow the directions on the screen.

Linux

If Flare was installed from a rpm package, it can be uninstalled with one of the following command lines, depending on whether it was installed with the `rpm` or `yum` command:

```
rpm -e cresset-flare-<vers>-1.<arch>
```

or

```
yum erase cresset-flare-<vers>-1.<arch>
```

If Flare was installed by unpacking a tgz file, it will be sufficient to remove the `Flare` folder from the disk.

macOS

In the Finder sidebar, click Applications. Drag the Flare icon from the Applications folder to the Trash (located at the end of the Dock).

Known Issues and Bugs

- Flare is not notarized for macOS: we recommend Mac users to check the user operations instructions to investigate how to operate with non-notarized software
- Flare may not operate optimally on remote systems like Citrix Virtual Apps. For optimal user experience, it is recommended to run Flare locally
- Optimal performance for GPU calculations in Flare is achieved with NVIDIA CUDA: please refer NVIDIA CUDA [instructions](#) for properly set up calculations using this framework. OpenCL is generally slightly less performant (~2-5%) but does not require any additional set-up
- Dynamics and FEP calculations will not run when using CUDA on NVIDIA graphics cards using the Turing microarchitecture or any successor architectures. It is recommended to use OpenCL as an alternative to CUDA
- In some occasions, Flare may experience a critical error on laptops with a Wacom pressure-sensitive pen and touch screen. Disabling the Wacom pen driver solves the issue.
- In some occasions, covalent docking of ligands containing a cis amide covalent warhead may generate unrealistic poses. Flipping the amide group in the input to the trans isomer fixes the problem
- In some occasions, covalent docking may fail to generate the correct stereoisomer for the docked ligands

- Flare will crash when the File dialog is opened if "Dell Backup and Recovery Application" between versions 1.7.5.64 and 1.9.2.8 (list not inclusive) is used on your computer
- In some occasions, the 3D window may not work smoothly in the Windows, Linux and macOS environment when using very large desktops. The workaround for this issue is to set the environment variable `QT_AUTO_SCREEN_SCALE_FACTOR=0`
- In some occasions, the active stereo (Quad Buffer) on Windows 10 may not work on older graphic cards, due to limitations of the NVIDIA graphic driver

Found a bug or problem that is not listed? Please [contact us](#) and help improve our software!

Third party software

A range of open source and free software libraries and executables are incorporated in each Flare installation. License agreement information for each third party software can be found in the 'documentation' subdirectory of the Flare installation directory and at <https://www.cresset-group.com/products/third-party-licenses/>. Contact us at support@cresset-group.com if you wish to obtain the source code for any open source component.

Support

Cresset
New Cambridge House
Bassingbourn Road
Litlington
Cambridgeshire, SG8 0SS, UK

+44 (0)1223 858890
support@cresset-group.com
www.cresset-group.com
Twitter: [@cressetgroup](https://twitter.com/cressetgroup)
LinkedIn: [Cresset group](https://www.linkedin.com/company/cresset-group)