



L I G H T W A V E<sup>®</sup>  
C O R E

**NEWTEK LIGHTWAVE 3D<sup>®</sup> WITH CORE TECHNOLOGY**  
**Features Document**

**Next Generation in 3D | NewTek Inc.**

**Q4 2009**



Model by Antanas Skucas (Cower) in LightWave 3D

**Full Disclosure: this is a forward-looking document.** It accurately represents the intentions of NewTek in the development of LightWave 3D® with CORE technology. (LightWave with CORE technology is a working title and not the final brand.) The software development process can be difficult to predict with certainty until delivery of final product. CORE consists of several technologies from external sources which are always under development or review; these technologies may or may not shape the final LightWave 3D with CORE technology feature list. **This document and its contents are subject to change without notice.**

LightWave and LightWave 3D are registered trademarks of NewTek, Inc.

LightWave CORE, FxMonkey, HyperVoxels, Intelligentities, Motion Mixer, MultiMeshes, Particle FX, P.A.V.L.O.V., Skelegons, SkyTracer 2, Vmaps, VPR are trademarks of NewTek, Inc.

## TABLE OF CONTENTS

LightWave 3D® with CORE Technology	4	Lighting	16
Here's what CORE does for you.	4	Animation	16
CORE Deliverables	4	Constraints	16
Direct Mesh Interchange	4	Point	16
MDD	5	Orientation	16
Other Assets	5	Scale	16
CORE User Interface Environment	5	Aim	16
CORE Workspaces	5	Nail	16
Model	7	Hinge	16
Surface	7	Cone Twist	17
Setup	7	Rendering	17
Animate	8	VPR	17
Render	8	CORE Rendering	17
Composite	8	CORE Color Management	17
3D Workspace Viewports	9	CORE Materials	17
Exploring CORE in More Detail	9	Dynamics	18
CORE uses Collada for Scene Files	9	Editors	18
Multiple Scene Documents/Referenced			
Scenes	9	Appendix 2: CORE SDK	19
CORE Preferences	10	Implications of the CORE SDK	19
Themes, Icons and Shortcuts	10		
CORE Input Device Support	10	Appendix 3: CORE Design/Architecture	19
CORE Editors and Panels	10	CORE Concepts	19
Modifier Stack	12	Industry-Standard Everywhere	20
Layer Panel	12	Development Platforms	20
Connection Editor	12		
Attribute Panel	13		
Outline Panel	13		
Image Editor	13		
Toolboxes	13		
Command History/Scripting	14		
Expressions	14		
All Wrapped Up	14		
Appendix 1: LightWave 3D with CORE Technology—			
Features and Capabilities	15		
File Interchange	15		
Selection	15		
Transform	15		
Align	15		
Model Construction	15		
Deformers	16		
Snapping	16		

## NewTek LightWave 3D® with CORE Technology

CORE is the next generation development foundation for 3D products at NewTek and one of the most aggressive new technologies in the industry. LightWave 3D® with CORE technology delivers the familiar tools and workflow functionality of LightWave®, with the flexibility to continue improving performance and feature sets for many years to come. The upcoming release of LightWave 3D with CORE technology allows you to work within LightWave Layout and Modeler and have access to the new benefits provided in CORE, many of which fulfill longstanding requests from the LightWave community.

### *Here's what CORE does for you.*

CORE technology significantly enhances the LightWave user experience with many high-level capabilities such as:

- **Parametric Modeling** - offers the freedom to change model attributes allowing you to explore options and make modifications at any stage in the creative process. All of your actions are preserved and updated using the modifier stack.
- **Interactive Viewport Preview Rendering (VPR™)** - any viewport can be switched to a VPR view, including multiple viewports with different angles. Based on the CORE render engine, all VPR views match the final CORE render output and are fully interactive, allowing you to create new models, modify existing ones, and interact with lights and cameras in the VPR viewport.
- **Rendering** - the CORE renderer, a subset of the LightWave renderer, is fully multi-threaded and leverages modern light transports, materials descriptions via CORE shaders, and global illumination.
- **Integrated Bullet Physics library** - offers fast, robust modern dynamics and simulation.
- **Animation of All CORE Attributes and Modeling Functions** - animate any data element offered in CORE, including direct animation of points, edges, faces and items (objects). Keyframe animation and procedural animation are offered.
- **Python-Based Expressions and Scripting** - a fully scriptable environment via Python and the CORE SDK can be used to create expressions and perform math operations in any data field. It can also be linked to the rotation of an object to change the color of another object.

- **Simple Compositing Environment** - good for blending layers, adjusting image sequences, and more. Supports plug-in products that conform to the OFX (OpenFX) standard proposed by The Foundry.
- **Fully Accessible SDK** - the completely open SDK gives full access to CORE to extend the out-of-box functionality of CORE.

## CORE Deliverables

The upcoming release of LightWave 3D will include CORE technology along with several new features in updated versions of Layout and Modeler with hooks for CORE. The Layout and Modeler applications include the improvements found in LightWave v9.6.1. Collectively, these new versions of Layout and Modeler are referred to as LightWave HC in this document.

NewTek recognizes that users will want to be able to continue working and earning a living while the CORE application progresses, so LightWave HC has been created to facilitate and provide an interchange with data created or edited within the CORE application. Several options for this are:

### **Direct Mesh Interchange**

This interchange allows objects to be moved back and forth between CORE, Layout and Modeler (if desired). Previous versions of LightWave facilitated this interchange with the HUB application. While the HUB will be shipped with LightWave HC, the HUB is not used with CORE, and it is recommended that users disable the HUB while working with CORE. Basically, the desired mesh asset is opened in each application, and once saved in an application, automatically updated in subsequent applications.

Direct Mesh Interchange works with:

- Mesh data
- Basic surfacing information
- Basic LightWave material channels

The following are not supported by Direct Mesh Interchange:

- LightWave nodes
- LightWave plug-ins (NewTek or third party)

## **MDD**

CORE features animation and it is necessary to facilitate the transfer of animation data into the LightWave Layout application. Since the motion system for CORE differs from LightWave Layout, there is no way to directly interchange animation information.

The interchange of animation data was resolved by NewTek long ago, through the creation of the animation interchange format, MDD. Most 3D animation applications today support the MDD format. MDD is native to LightWave Layout and is the perfect solution to allow CORE animation data to be used within the LightWave Layout application.

## **Other Assets**

CORE can handle the interchange of texture maps and images, however it cannot use procedural textures and shaders from LightWave Layout. Since the underlying architecture of CORE is completely different than the original LightWave architecture, plug-in products written for LightWave will not operate in CORE. They will, however, continue to operate in LightWave HC, with one caveat for the Macintosh platform: all Macintosh LightWave HC plug-ins must be written for Cocoa, not Carbon.

The approach NewTek has taken to LightWave with CORE technology offers tremendous flexibility for the user. You may work exclusively in LightWave HC with its familiar environment but you have the option to tap into the additional power and benefits offered in the CORE application.

## **CORE User Interface Environment**

There are notable differences between the CORE, Layout and Modeler applications. CORE is an integrated application, one that allows you to model, animate and render within the same environment. And, like Layout and Modeler, you will see 3D viewports, toolboxes and so on. These things will work in a similar way to Layout and Modeler, but there will be differences too—often, these differences are found in the integration of modeling and animation.

Some differences are immediately apparent once you begin to work with CORE—many of which fulfill longstanding requests from the LightWave community such as: the ability to freely move panels around and have them dock together, and more. In fact, the user interface has several

skins or themes to choose from, or if you prefer, you can even make your own. It is all CSS-based, so you can easily edit the CSS file with any text editor.

Once you are familiar with CORE, you can choose what you need to do in a particular session. CORE has many tools for creation and problem solving. However, instead of trying to fit all of those tools into huge menus and endless toolboxes, we opted to categorize specific tools into workspaces.

The CORE user interface environment consists of specific workspaces; each workspace consists of several different panels and viewports. Each of these toolboxes can be “torn off” and placed wherever you desire. Tabbed panels can be made into freestanding items, or docked into other areas of the CORE interface.

Workspaces, discussed in the following section, set the stage for what you want to do. But in order to adequately present that material, it is necessary to understand the benefits of the CORE tools workspaces, which are covered in this section.

## **CORE Workspaces**

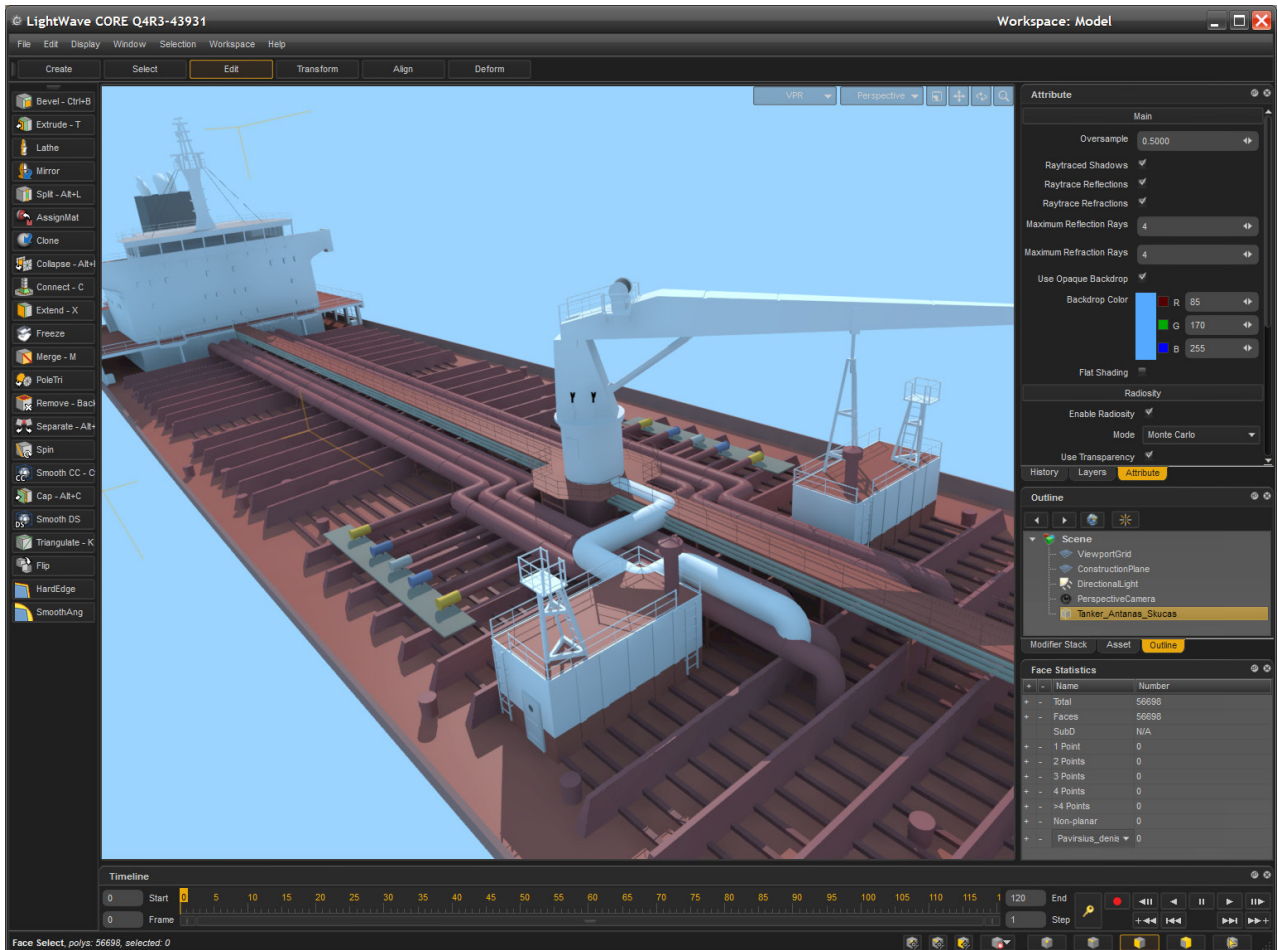
The content creation process can be very involved. In modern applications, there are so many different tools and methods to utilize, depending upon the task at hand. Fortunately, these tasks can be categorized and presented in a way that aids, rather than impedes, the creative process. CORE does so with a series of user interface and toolbox layouts called workspaces. (These workspaces are not locked; artists are free to access any toolset or panel at any time via the application menu.)

The workspace concept keeps the most needed tools close at hand, while minimizing the tools that are used less often.

In addition to the factory workspaces offered in CORE, you can tailor and save your own versions for future use.

CORE offers the following workspaces:

- Model
- Surface
- Setup
- Animate
- Render
- Composite



CORE Model Workspace (Thor theme) Model created by Antanas Skucas (Cower) in LightWave 3D

## Model

The Model workspace is essentially a “model shop.” Toolboxes used to create 3D models are located in the Model workspace, the default workspace in CORE. The Model workspace contains a variety of different toolboxes for creating and modifying 3D meshes.

Toolboxes with factory settings are found in the tabs across the top of the CORE interface. Toolboxes, like other aspects of CORE, can be reoriented and repositioned as needed. Toolboxes are presented as buttons or tabs. Selecting any of the toolboxes will show the available tools within the toolbox in a separate dock. You can place a toolbox anywhere on your desktop by dragging to the desired location.

- **Create** – The Create Toolbox is for creating the basic building blocks of a mesh. A variety of tool primitives are available (see Appendix 1). These primitives can be further edited into more complicated shapes as needed.
- **Select** – The Select Toolbox contains tools for the process of selecting elements of a mesh, or whole meshes; a selection is necessary for tools to operate. As meshes can be very complex, CORE provides several specific selection tools to fine tune the selections to be used.
- **Edit** – The Edit Toolbox is used to further modify the meshes in a scene. Typically, you would start with primitives and then use the tools in the Edit toolbox to refine the shape of scene meshes.
- **Multiply** – The Multiply Toolbox serves a similar purpose as the Multiply tab in Modeler, in that you are typically adding to an existing mesh dataset with the tools from in this toolbox.
- **Transform** – The Transform Toolbox gives you tools for precisely placing meshes within a scene. Placement of a mesh requires the ability to move in 3D space, orient the mesh as needed, and scale the mesh to the proper size.
- **Align** – The Align Toolbox assists in placing and aligning scene meshes and other scene items such as cameras and lights.
- **Deform** – The Deform Toolbox contains tools that further change the shape of scene meshes in a volumetric fashion. These tools capture regions of a mesh, and further change the mesh based upon the tool type.

The Model workspace may be one of the most used workspaces in CORE. Objects are created and edited in this workspace using the tools; background images can be added to the scene to further assist the modeling process.

## Surface

The Surface workspace is like a paint shop or graphics department. It is in this workspace that various tools and viewports are presented to help you surface scene meshes. “Surfacing” refers to creating the “look” of an object and involves the assignment of colors, textures, material types, and so on to a scene mesh, or portions of a scene mesh.

When you choose the Surface workspace, the UV editor and image editors appear and a 3D view is presented, as well as the Asset panel, Attribute panel, and UV Edit toolbox.

- **UV Edit** – The UV Edit toolbox is used, along with the UV Editor, to create and edit UV meshes. UV coordinates, also known as “texture space” coordinates, are needed to provide images with the information to properly conform to the topology of a given mesh. In order to achieve this, a mesh must first be “flattened” into a space that the image can then work with—similar to cutting a soap box to lay flat on a desktop. Several tools are included to help facilitate the process.

## Setup

The Setup workspace is where artists and technical directors create and set up character rigs and rig controls. Dynamics attributes and simulations are also managed in the Setup workspace.

The following user interface items and toolboxes are part of the Setup workspace:

- Timeline
- Deformation
- Dynamics
- Joints
- Constraints
- Scripting

## Animate

CORE was designed from the ground up to animate any attribute within a scene. Model operations, including the ability to animate portions of items, such as points or selection sets, for example, can be animated over time. The same is true for lighting and cameras.

All of the attributes of any scene item can be animated, associated, referenced and so on through a variety of methods, including:

- Keyframe Animation
- Procedural Animation
- Scripted/Expression-Based Animation

Any attribute can be animated, and via expressions (the Connection Editor and other methods), complex relationships can be created and animated.

Almost anything can be animated using CORE. For example, if you need to create an animation of a running engine with the internal mechanisms in motion, you can use Boolean operations to cut away aspects of the engine to reveal the cam shaft, pistons, drive train, and so on. As engines tend to be complex, the Boolean subtract operator can be animated to different areas of the engine at different key points in time as determined by the script or storyboard.

The Animate workspace is where you animate your work over time. When you choose the Animate workspace, the user interface reconfigures to present 3D viewports, a scene timeline, an outline panel, an attribute panel, the Select Toolbox, and so on.

In addition to the above, the following items are also presented as aspects of the Animate workspace:

- Creation Tools
- Deformation (Subset)
- Transforms
- Dynamics (Collision Detection, Etc.)
- Constraints
- Scripting

Managing keyframes and animation can be accomplished via the timeline graph editor. Animation is stored in motion curves, expressions, dynamics and/or procedural data for each item or item element flagged for animation. Animation rotation data can be expressed via Euler angles or quaternions with independent axes.

Skeleton (joint) animation and inverse kinematics are also available, though there is no skinning method in the initial version of CORE; it is for this reason that the term “basic” animation has been used to describe animation offerings in this version of CORE.

## Render

The Render workspace is where artists and technical directors prepare the scene for final rendering. For high-end 3D applications the rendering process can be very involved, so CORE presents a dedicated workspace to make this process easier. The Render workspace consists of the following items:

- Render Passes
- Render Regions
- Render Config Tools
- Render Management

## Composite

The Composite workspace is a simple 3D compositing environment where you can create and manipulate texture maps and rendered imagery by compositing elements to create compelling final effects. While not as sophisticated as a dedicated compositing package, the Composite workspace in CORE allows you to move to a dedicated compositing application as necessary.

When you choose the Composite workspace, the Image Viewer appears, as does the Connection Editor. A small Perspective/Camera view of the active scene is also present. The timeline runs along the bottom of the UI so composites and texture treatments can be properly managed over a period of time.

## OFX Plug-in Support

The Composite workspace in CORE is capable of image manipulation and compositing via a selection of internal tools, as well as available third-party plug-ins that support the OpenFX standard (OFX). OFX is an open source plug-in standard for developing 2D digital visual effects. OFX is recognized throughout the industry not only for its benefits to digital artists but also for allowing developers to spend more time focusing on R&D to create tools for compelling digital effects.



The Foundry (<http://www.thefoundry.co.uk/>), Re:Vision Effects (<http://revisionfx.com/>) and GenArts (<http://www.genarts.com/product/sapphire/ofx/fxlist>) and several other prolific plug-in vendors offer a variety of image processing and manipulation plug-ins in OFX format, such as keyers, motion blur products and so on. The products can be used to create better texture-mapped imagery, motion input imagery, and rendered output from CORE.

The Composite workspace does not supply a keyer node (used to extract mattes from pre-existing footage, typically blue or green screen live action or miniature photography). This task can be accomplished with an OFX plug-in such as Keylight from The Foundry<sup>1</sup> directly within the CORE composite workspace.

As the list of OFX compliant products grow, so does the ability of the Composite workspace in CORE.

*Note: As these products are intended for professional use, they are priced accordingly.*

The Composite workspace is a node-based compositing environment that will properly layer and combine images that contain embedded mattes or “alpha channels.” The Image Mixer node allows for two inputs and a mix input, which controls the degree of blending between the source images. If images have alpha channels, there is no need to further drive the mix value, as the images will be properly layered on top of each other at full value as dictated by the values in the alpha channel matte.

Final output from the composite depends upon the needs of the image processing or shot. As the CORE composite workspace is a 3D environment as well as a 2D or 2.5D environment, many interesting effects are possible. In addition, you can determine if final output is rendered via OpenGL or using the CORE rendering engine for greater flexibility. OFX plug-in support allows the extension of compositing capabilities within LightWave CORE. If you have access to these types of plug-ins you have the ability to leverage your investment in OFX products across applications that adhere to the OFX standard. Different versions of these products are not necessary for applications that support the OFX standard. Artists using dedicated compositing applications that support this standard are likely to have OFX plug-ins already, and these plug-ins can be used within the CORE Composite workspace as well.

### **3D Workspace Viewports**

In all of the CORE workspaces, 3D viewports are always present. These viewports can be further modified as needed. Each viewport can be toggled as the primary viewport. Viewports can have independent shading methods applied to suit your needs.

Each and every viewport has the ability to be viewed as either an OpenGL viewport or as a VPR viewport. Viewports can be “torn off” the main application window and placed on other screens within the desktop.

### **Exploring CORE in More Detail**

Earlier we discussed CORE deliverables, the CORE user interface environment, and CORE workspaces. Now it’s time for a brief discussion of scene files, and a look at the editors and panels in CORE.

### **CORE uses Collada for Scene Files**

CORE introduces a new scene file format, a result of merging the capabilities of modeling and animation into a single application. This new scene format is based upon the industry-standard Collada format. We chose this format to enhance a more seamless integration of LightWave CORE into non-LightWave-based pipelines. We also chose this format because the Collada format offers all the structures a combined application needs.

### **Multiple Scene Documents/Referenced Scenes**

One significant improvement that CORE offers is the support for multi-scene document editing and asset management. You can choose to work with these multiple scenes as different shots or variations of the same shot. Multiple scenes are presented and managed in the outline panel.

In addition to multi-scene editing in the same session, scenes can be referenced within other scenes, allowing you to break tasks into more manageable documents. Doing so will allow you to focus on the purpose of that particular element. When the time comes, you can also reference that scene in a larger master scene.

<sup>1</sup>The Foundry requires that OFX host applications be certified by the company. At the time of this writing, LightWave CORE has not yet been submitted for certification, as it is still under development.

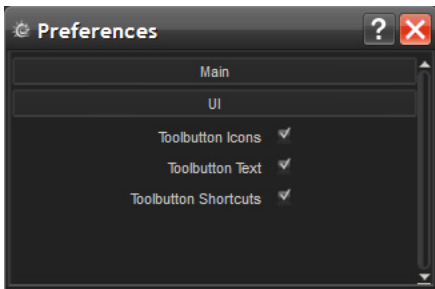
## CORE Preferences

To select the CORE environment most suitable to your style of work you will want to visit the preferences section. Here, you can choose the LightWave v9 keyboard shortcut presets, the factory presets, or create your own. The following describes these features in further detail.

## Themes, Icons and Shortcuts

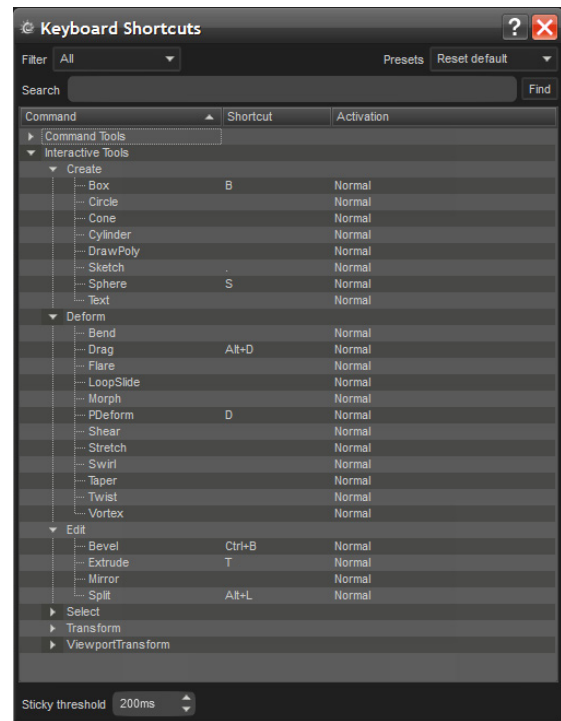
You will notice in the following sections a variety of different appearances for CORE. These themes are factory presets, but you can easily create your own since the themes are in the industry-standard CSS format.

In addition to themes, CORE introduces something new to the LightWave lexicon: icons. Many users prefer text-only buttons—LightWave with CORE technology gives you both. You can choose whether or not to display text buttons only, icons only, or both.



Preferences Panel and Button preferences

Of course, shortcuts are vital to any good workflow, and CORE offers many options. The Keyboard Editor allows you to set the shortcuts you like and also choose from interaction models and factory defaults.



Keyboard Shortcuts

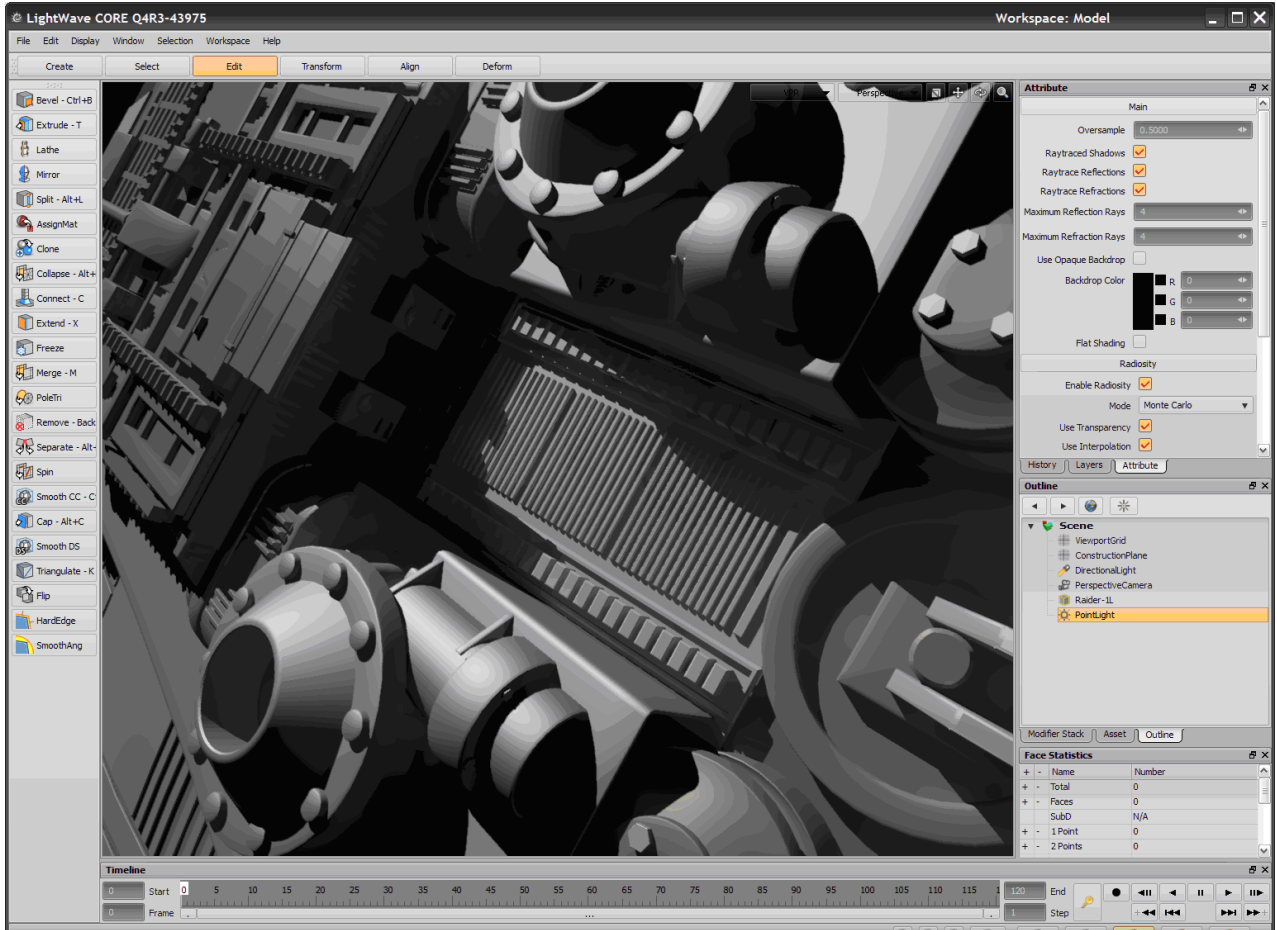
In addition, CORE offers a number of presets to make working with CORE that much easier.

## CORE Input Device Support

The mouse is not always the preferred method for creating 3D. CORE is designed to be very “tablet friendly,” and supports tablets from companies such as Wacom and input devices like the Space Pilot and Space Traveler from 3D Connexion. If your favorite input device is not mentioned here, the CORE SDK makes adding devices straightforward for vendors of these products.

## CORE Editors and Panels

CORE introduces many new editors and panels in its toolbox. Each of these editors and panels has a specific function and purpose. Some editors are present when you start up the application, while others are available from menus. The following sections discuss each of these editors and panels in detail.



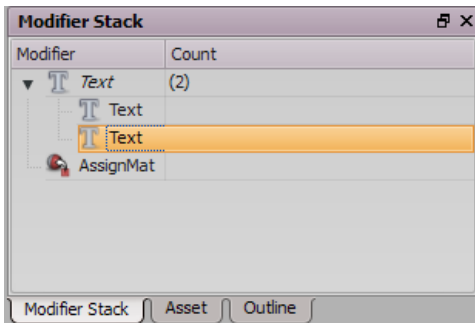
The CORE UI components for the Model workspace (Sumatra theme). Modeled by Jim Tally in LightWave 3D

## Modifier Stack

The CORE modifier stack is an attribute of each mesh item. Every CORE mesh item has a modifier stack behind it. The modifier stack is a living record of all of the geometric operations applied to a specific object. Operators in the stack can be rearranged (doing so can produce notably different results), enabled, frozen (so as to be un-editable or “flattened”), muted and deleted on command.

The modifier stack is available in the modifier stack tab for any selected geometric object in CORE. The modifier stack is saved with an object each time it is saved.

Some people may prefer not to work with the modifier stack. You can use or ignore the modifier stack as necessary.

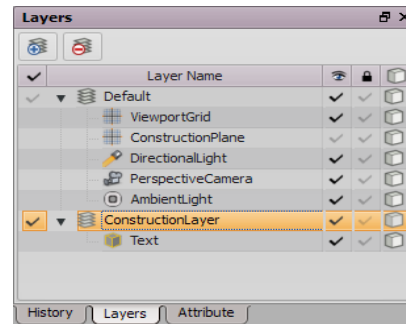


The Modifier Stack

## Layer Panel

The layer panel is different than the layer panel in LightWave Modeler because this panel is used for modeling, animation and general layout tasks. This panel is where you create layers to place your scene items. With layers you can:

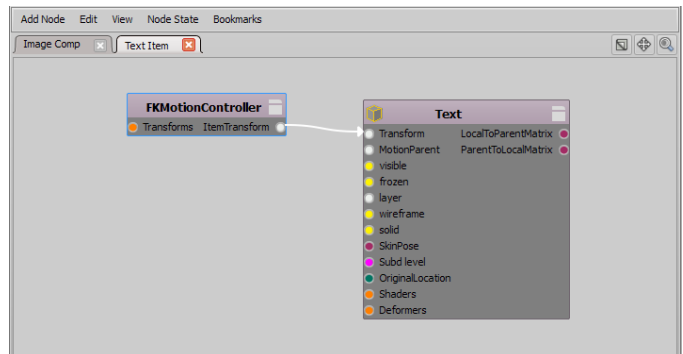
- Hide objects from view
- Lock objects, to prevent accidental selection
- Have unique drawing styles with layer content
- Keep the workspace better organized



Layers Panel

## Connection Editor

The Connection Editor allows you to create interconnections and relationships with all aspects of a CORE scene. These relationships can be as elaborate or as simple as the scene requires. For example, you may wish to have the segments of an object change with the distance of the object from the viewer. This can be done with either the connection editor, or the expression system (they are effectively the same, and are interoperable).



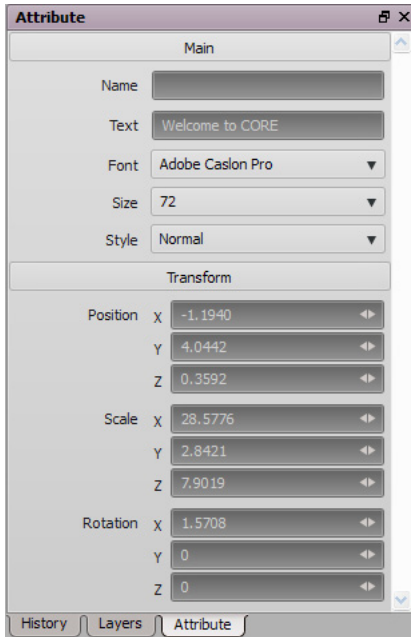
Connection Editor

Items are dragged from the outline panel into the Connection Editor for editing. You can add nodes and various operators to create expressions, modify attributes, and so on. Bookmarks are available within the Connection Editor to help keep things organized, and you can copy and paste a node reference when you need to have subsections of node flows on other bookmarked pages.

In addition, grouped nodes help reduce node-flow clutter when the flows are combined.

## Attribute Panel

The CORE attribute panel is used to manage numeric and attribute data of CORE objects, set CORE tool options, and so on. It helps keep UI clutter to a minimum. The CORE attribute panel contents will change depending on the object or tool selected.



The Attribute Panel

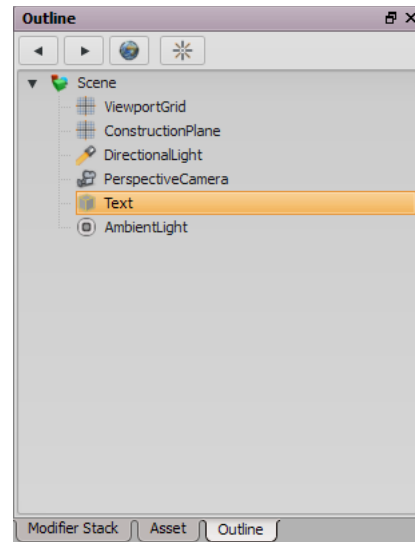
The attribute panel is one of the most important panels in CORE, as it is the primary focus of numeric and data entry in CORE, outside of the Command History panel.

The attribute panel contents change dynamically depending upon the tool or item selected. VPR settings are also adjusted via this panel.

## Outline Panel

The outline panel is similar to the Scene Editor in LightWave v9.6.1, however, animation channels are not presented in this panel. This panel is used to manage scene items, make item selections, create new scene items, organize hierarchy, and more. You can also drag external assets into CORE from the desktop and outline panel.

Additionally, you can drag and drop items together to form hierarchies, or drag items from the outline panel to the Connection Editor to build relationships and node flows.



## Image Editor

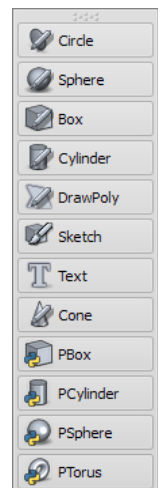
The Image Editor allows you to view images loaded into CORE, and offers basic image editing functions (most editing functions are found in the Connection Editor). This viewer conforms to the CORE color management system, and will allow you to choose the appropriate gamma and LUT settings for scenes and pipelines.

Images can be loaded directly into the image editor by double clicking in a null space within the editor, or by choosing File>Add from the main CORE application menu. When images are added to the scene, they will also appear in the Asset panel.

For a list of supported image types, please see Appendix 1 – LightWave 3D with CORE Technology Features and Capabilities.

## Toolboxes

Toolboxes are the heart of the CORE toolset; several toolboxes are available—all targeted at specific areas of interest. Toolboxes are available to screen left, but can be placed anywhere in the user interface.



Tool Box Example

## ***Command History/Scripting***

CORE outputs all of its commands to a Command History panel. These commands are presented in a Python-compatible format, allowing commands to serve as a basis for user-written scripts. Using scripts to automate processes is a powerful concept, and the Command History panel is a great way to get started.

Industry-standard Python forms the basis of the scripting in CORE (currently version 2.6). The Python implementation is layered into CORE via SWIG. SWIG is a language-interfacing layer that allows the CORE SDK to be accessed through languages other than the factory Python language that ships with CORE. For example, a game development company may wish to use Lua instead of Python to interface with the CORE SDK. SWIG supports this (though the company would need to use the SWIG language interface generator to expose the CORE SDK to the language of choice).

## ***Expressions***

Expressions in CORE are Python-based. Any text field can create expressions, and link to any CORE attribute. Any attribute in the application can be used for associations and expressions, as well as math operations. When an expression is present on any CORE attribute, the attribute will change color to indicate the type of expression or constraint being applied.

## **All Wrapped Up**

LightWave 3D with CORE technology represents the most modern and aggressive application development for 3D content creation. The following appendices present information on specific features, a brief note on the CORE SDK, and a discussion on design and architectural considerations. As always, this information is subject to change without notice before final product shipment.

We hope you found this document and its information useful. For more updates, please visit the NewTek Web site [www.newtek.com](http://www.newtek.com), as information will be periodically updated. To be the first to receive LightWave 3D with CORE technology development information, as well as immediate access to the latest builds, [join the HardCORE™ membership program today.](#)

We look forward to providing you tools to support you as a leader in the content creation industry and to have you as a member of the NewTek LightWave family now and for many years to come.

## Appendix 1: LightWave 3D with CORE Technology— Features and Capabilities

### ***File Interchange***

- 3D Formats
  - LWO
  - Collada
  - FBX
  - MDD
- Image Formats
  - TIFF
  - TGA
  - JPG
  - PNG
  - OpenEXR
  - IFF
  - SGI Raw

### ***Selection***

- Marquee
- Lasso
- Select
- Contract Selection
- Expand Selection
- Invert Selection
- Select By (Material)
- Select Connected
- Select Loop
- Select Outline
- Select Ring
- Selection Sets

### ***Transform***

- Scale
- Translate
- Rotate

### ***Align***

- Align to Center
- Align to Ground
- Align to Construction Plane

### ***Model Construction***

- Boolean Operations
  - Union
  - Subtraction
  - Intersection
- UV Editing
  - UV Relax
    - UV Pinning
  - UV Stitch

- UV Tiles
- UV Unwrapping
- Multiple UV Maps
- Construction Planes
- Scene Units
- Vmaps and Map Editing
- Brush Sculpting
- Create
  - Curve
  - Circle
  - Sphere
  - Box
  - Cylinder
  - Draw Polygon
  - Sketch
  - Text
  - Polygon Box
  - Polygon Cylinder
  - Polygon Sphere
  - Polygon Torus
- Edit
  - Bevel
  - Extrude
  - Mirror
  - Split
  - Assign Material
  - Collapse
  - Connect
  - Extend
  - Freeze
  - Merge
  - Pole Triangulate
  - Remove (Delete)
  - Spin (Element)
  - Smooth with Catmull Clark
  - Smooth with Doo Sabin
  - Cap (Holes)
  - Triangulate
  - Flip (Normals)
  - Hard Edge
  - Smoothing Angle
- Multiply
  - Array
  - Booleans
  - Clone
  - Lathe
  - Separate

## **Deformers**

- Bend
- Drag
- Flare
- Lattice
- Loop Slide
- Morph
- Paint Deform
- Shear
- Stretch
- Swirl
- Taper
- Twist
- Vortex
- Flatten

## **Snapping**

- Point Snap
- Edge Snap
- Face Snap
- Grid Snap
- Angle Snap

## **Lighting**

- Direct Light
- Ambient
- Radial Light
- Spot Light
- Area Light
- IES Light Support
- HDRI/Image-based Lighting

## **Animation**

- Item Transforms (Move, Rotate, Scale)
- Element (Sub Item) Transforms (Move, Rotate, Scale)
- Modeling Operations
- Set Keyframe
- Autokey
- Delete Keyframe
- Graph Editor

## **Constraints**

### **Point**

Point constraints limit an object's movement. They are used to attach the location of one object to another without parenting, to provide more flexibility. For example, you can use a point constraint to lock the mesh of a container to the model of a truck. Then at the loading dock, you could animate the constraint to transfer over to a forklift, lifting the container from the truck and placing it in storage. This is accomplished by keying the weights of the constraints for the truck and the forklift with the proper timing for the animated shot.

### **Orientation**

Orientation constraints affect the rotation of an object. Orientation constraints are great when you need to match the orientation of one object to another, such as the wheels of a car in motion. You would have the front wheels constrained in one set, and the rear wheels constrained in another. When the front left wheel turns on its axis, the right wheel would match. And, when the car moves in any direction, the rear wheels (without steering) would work correctly if an expression to automatically rotate the wheels, based on distance, is added.

### **Scale**

Scale constraints are used when you need to make the scale of one object match the scale of another – either for the entire length of a shot or for a portion thereof (as this can be animated in strength).

### **Aim**

Aim constraints are great for targeting. An aim constraint will allow the constrained object to track another object, orienting the constrained object as necessary. Think Galactica cannons shooting Cylons out of the sky.

### **Nail**

Nail constrains a rigid body to a point in world space.

### **Hinge**

Hinge constraints limit motion to one axis, like the hinges on a gate. You can also use them to simulate the connection of a tire to a car.



## Cone Twist

The cone twist constraint comes from the Bullet library, and is used for dynamics only. It is useful for creating rag doll-like effects, as is a good tool to use when simulating the upper arm of a human.

## Rendering

LightWave 3D with CORE technology offers two different flavors of rendering: Viewport Preview Rendering (VPR) and CORE rendering.

### VPR

The CORE viewport rendering engine allows for direct interaction with elements in the rendering, from the rendered view. Any viewport can switch to VPR, and even portions of viewports can be specified for viewport rendering.

Other applications do offer pre-visualization in a similar manner, but none offer the ability to directly interact with data, including creating new meshes, within the rendered viewport.

VPR is essentially the CORE rendering engine operating in an iterative, interactive fashion. Whatever CORE can render, VPR can also render. The results will be virtually identical, with some exceptions, as VPR does prioritize to deliver on speed and interactivity.

As VPR is a complete version of the CORE rendering engine, not a subset, it can render global illumination interactively, however, with less interactivity due to the amount of performance required to produce GI renderings. Fortunately, the CORE rendering engine is fully multithreaded, another benefit leveraged by VPR. The more processors you have available to you, the faster VPR will render.

VPR is NOT a GPU-based renderer. However, when CORE leverages GPUs for more rendering, VPR will too because it shares the same codebase.

## CORE Rendering

The upcoming version of LightWave 3D with CORE technology will have a subset of the rendering capabilities of LightWave v9.6.1, although the underlying architectures are different.

The CORE renderer is fully multi-threaded, and leverages modern light transports, modern materials descriptions via CORE shaders, and global illumination.

## CORE Color Management

The CORE rendering workspace is linear, with LUTs for viewers and color pickers. This helps to ensure the best result when working with texture maps and rendered images, as well as seeing the best results with light falloffs and global illumination settings. Linear workflows are the standard for 3D rendering today in production pipelines for film, television and print.

## CORE Materials

CORE Materials come in two flavors: traditional LightWave materials, to better facilitate interchange with LightWave HC, and Collada materials, to better facilitate interchange with other applications.

### Collada

- Blinn
- Phong
- Constant
- Lambert

## Textures and Procedural Generators

A variety of image map formats are available for textures (see File IO), and via the Composite State, these images can be processed in a variety of ways to fit the needs of the scene. You can choose from the available UV projections that have been created for the mesh upon which to apply the texture images. Textures can be used to create more complex materials as needed, whether to modulate diffuse values, or to act as masks for other shading effects.

Procedural Generators are also available to further fulfill your material requirements. These generators are animatable, and can be connected with other attributes available within a CORE scene. The following CORE Procedural Generators are currently available:

- Fresnel Dielectric
- Voroni 3D Cell
- Turbulence Generator
- Perlin Noise 3D Generator

## **Dynamics**

CORE integrates the popular Bullet Physics Library (<http://bulletphysics.org/wordpress/>) into the CORE unified dynamics solver. Bullet supports both rigid body and soft body collisions. Bullet plans to offer OpenCL support in the future, which fits with the development of CORE.

The Bullet Physics Library offers the following benefits:

- Open source C++ code under Zlib license and free for any commercial use on all platforms including PLAYSTATION 3, XBox 360, Wii, PC, Linux, Mac OSX and iPhone.
- Discrete and continuous collision detection, including ray and convex sweep test. Collision shapes include concave and convex meshes and all basic primitives.
- Fast and stable rigid body dynamics constraint solver, vehicle dynamics, character controller and slider, hinge, generic six degrees of freedom and cone-twist constraint for rag dolls.
- Soft Body dynamics for cloth, rope and deformable volumes with two-way interaction with rigid bodies, including constraint support.

In CORE, you can use dynamics for simulation, modeling, or interaction with scene items. Animation created in CORE can be exported into LightWave Layout (and other applications) via the MDD file format.

**Modeling with dynamics example:** Consider an alley scene with crates. You can choose to place the crates (and other debris) by hand, or you can choose to use the dynamics tools to help streamline the process.

## **Editors**

- Asset Manager
- Attribute Editor
- Command History
- Command Shell
- Connection Editor
- Curve Editor
- Graph Editor
- Hierarchy Editor
- History Stack
- Image Editor
- Keyboard Shortcuts Editor
- Log Window
- Modifier Stack
- Layers Panel
- Outline Editor
- Plug-in Manager
- Preferences Editor
- Presets Manager
- Text Editor
- Theme Manager
- Timeline
- Render Configuration Panel
- Web Browser

## Appendix 2: CORE SDK

The CORE SDK is full-featured and fully accessible. It is essentially the same SDK used to develop CORE technology. During the development phase for CORE, the SDK was expanded while the application was being developed; all tools will be migrated into the SDK. Future revisions of CORE will add to the SDK as well, and allow full access to all functions offered by CORE. Certain restrictions may be placed depending upon legal requirements of libraries used, such as Qt: we do not expose the entire Qt system through the CORE SDK, but instead provide our own SDK hooks into our license of Qt, as allowed by Nokia. Third-party developers should not require their own license of the Qt toolkit, unless they wish to completely control user interface elements directly, or wish to develop standalone applications. NewTek makes no warranties or guarantees about the above information, and recommends that all interested third-party developers consult with legal representation and review the Nokia Qt licensing agreements before proceeding with commercial development. Nokia has a blog that provides licensing answers for Qt at the link below:

<http://blog.qt.nokia.com/2009/11/30/qt-making-the-right-licensing-decision/>

The CORE SDK may be the most important aspect of CORE. No application is capable of being all things to all users, so it is imperative that an application be as extensible as possible, both to fulfill the needs of the user, and to future-proof the application. Modern production pipelines are typically script-driven to facilitate inter-application communication and data transfer, asset management, and so on. Without a full-featured SDK the ability of an application to integrate into a production pipeline would be limited.

The CORE SDK also facilitates third-party application development of products not necessarily offered by NewTek in a particular version of the software.

### ***Implications of the CORE SDK***

Because CORE is developed with the same SDK that it ships with, third-party developers can do everything the CORE development team can do with CORE technology. There is no restriction to other vendors adding in their own rendering engines or toolsets, CORE treats them all equally.

## Appendix 3: CORE Design/Architecture

LightWave 3D with CORE technology is a modern application with next generation technologies, built around industry-standard structures and libraries. Written from the ground up in C++, it is very efficient, both in speed and memory. In a 32-bit operating system, for example, CORE is three to five times more efficient in RAM usage than LightWave v9.6.1 or LightWave HC due to its streamlined architecture.

CORE is designed for modern 64-bit operating systems, but runs equally well in 32-bit operating systems running Windows 7, Windows Vista, Macintosh Snow Leopard or later, and various Linux releases.

CORE is multithreaded and multiprocessor aware. CORE leverages state-of-the-art graphics cards and the more processors you have, the faster your sessions with CORE will be. GPU subdivision surfaces in CORE, for example, are GPU aware. Other CORE functions will leverage the GPU in a variety of ways over the course of CORE development.

CORE is an integrated application, with modeling, rendering, simulation and animation functions operating within the same application. This approach is very different than LightWave v9.6.1 and LightWave HC, which offer separate Layout and Modeler applications. With the opportunity to use familiar LightWave skins and presets longtime LightWave users can set the new CORE environment up to best suit their needs.

The CORE application is node-based, and utilizes a scene graph database. This structure is very fast to interact with programmatically, and very versatile.

CORE also incorporates a modifier stack for modeling operations. This approach is considered more user friendly than an all node-based format, which can be found in other 3D applications.

### ***CORE Concepts***

CORE is object-oriented, parametric node-based architecture. Built on a scene graph, CORE represents all data within a scene as a node in a graph that emanates from a root node. In some cases, other ways of dealing with and storing data and operations are also used, such as the modifier stack. The modifier stack stores modeling operations for each mesh item within a scene. As a result, the CORE data flow is a hybrid node/modifier stack design.

## ***Industry-Standard Everywhere***

Throughout this document, you have seen frequent use of the term, “industry-standard.” One of the CORE design goals was to use industry-standard technology whenever possible instead of reinventing the wheel. Some of the reasons for this approach:

- Satisfying customer demand
- Future-proofing
- Interoperability with other applications
- Saves development and implementation time
- Faster integration of proven technology

## ***Development Platforms***

CORE is written in C++, and designed as an object-oriented application. CORE uses several different technologies and libraries to deliver state-of-the-art performance. For the user interface, CORE uses the Qt libraries from Nokia. The Qt environment is fast becoming a standard throughout the software development industry. It is available on all platforms, allowing for a single codebase development process, which is very important when you have more than one host operating system to support. By using Qt, we can offer LightWave CORE for Linux as well as Windows X and Mac OS.