# Rebuilding Kinsale: 1601

## *Historical re-creation using LW5.6 & WCS5*

By R Scott Cherba • Images by the Author

In previous World Construction Set (WCS) articles, I have introduced you to little pieces of projects to spark your imagination. We're going to make a great leap this time and consider a major WCS undertaking with the recreation of Kinsale, Ireland before the Battle of Kinsale in December 1601. We'll model a medieval town in LightWave 3D 5.6 and bring it into WCS5. The project will form the basis for stills and animations spanning several months of history in a documentary currently under production by Fir Bolg Films (**www.firbolgfilms.com**).

### Terrain, anyone?

Note: *What's a DEM? The term "digital elevation model" is a sweepingly generic term that has come to mean terrain data of any flavor. Within this article's context, it refers to WCS .elev elevation files.*

Before we can do anything, we'll need some terrain. Fortunately, an AutoCAD file containing 10-meter elevation contours and cultural features of the area was available for a nominal fee from the Ordnance Survey of Ireland (OSI). The file was edited in AutoCAD r14 to remove all data except the elevation polylines, and then exported as a DXF. The DXF was imported into WCS as Control Points, and it took a few days to grid the points into a terrain surface.

The resulting DEM was imported again to place the terrain at the correct latitude and longitude, tile the DEM, and add vertical exaggeration to double elevations. Scaling elevations upward may not accurately represent the terrain, but is often necessary in areas of low relief where wide-angle shots make the terrain appear flatter to the viewer. Figure 1 shows an overhead view of the terrain with the DEM grid superimposed. The grid Ground Effect is a modified version of the Green Acre Grid from 3D Nature's recently released *WCS Content CD*.

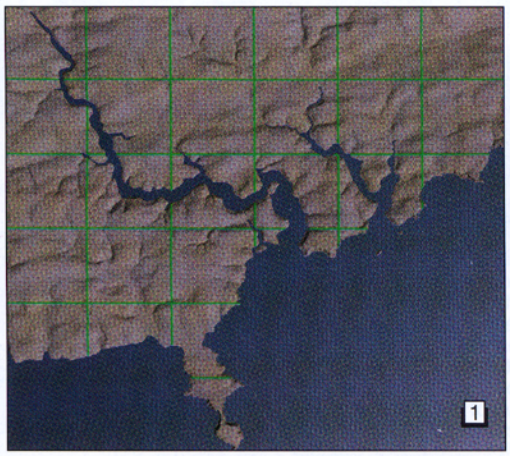
Figure 2 is a detail showing the area we'll be working with. The town of Kinsale is on the south coast of Ireland in county Cork, about 20km south of the town of Cork. It overlooks a protected bend in the harbor. Access to the inner harbor is guarded by fortifications with clear views of the outer harbor. For now, we're concerned with re-creating the town.

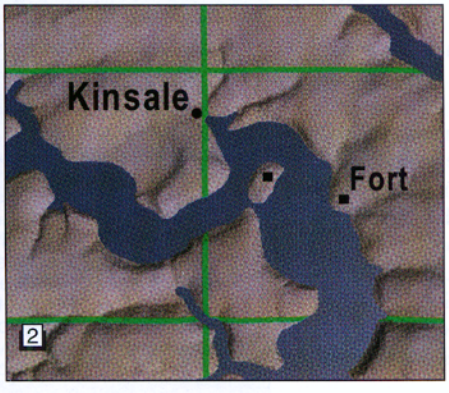
When it comes to 3D objects and WCS, we have three choices: 3D objects can be imported into WCS, a WCS scene can be exported as a LightWave or 3D Studio scene* for objects to be integrated there, or files can be rendered for scene assembly in a compositing program. For this project we're going to import LW objects into WCS. Why? We don't have a detailed storyboard for the film and we need to be able to quickly change scenes and render them as the producer makes changes. Working with 3D objects within WCS allows us to quickly make changes and render new frames.

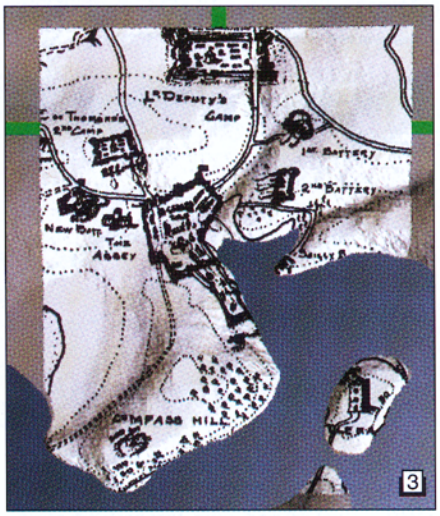
### Research!

Things have changed a bit in Kinsale in 400 years. Except for a few isolated buildings, nothing remains of the original town. The old town walls have long since fallen or been removed, and little remains to suggest where they were. The harbor forts were razed and new forts built over them. The dense oak woodlands are gone. Worse yet, records of the period and a reliable history don't seem to exist. One of the goals of the film is to use animated re-creations to help historians visualize what might have happened in December of 1601.

This is where research comes in. We have a map made by a researcher early last century, when remains of the original town wall still existed. Figure 3 shows the Kinsale section of the map draped over the terrain as a Color Map. The town itself is not much bigger than it was in 1601, and the downtown street layout has probably not changed much since then. We don't know what the town wall and gates looked like, but we have video footage, still photographs, and drawings of contemporary structures still existing in Ireland (Figure 4).





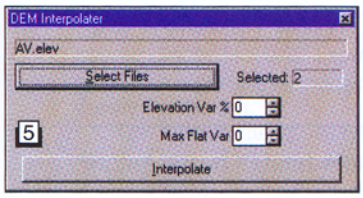
Courtesy Fir Bolg Films

Medieval Kinsale may be gone and almost forgotten, but we have enough information to re-create a reasonable facsimile.

Due to the number of polygons involved, the town must be modeled as several objects. The modeling method will ensure that the objects can be easily imported into WCS in their correct positions, which first will require some work in WCS to export a terrain object for reference in LW.

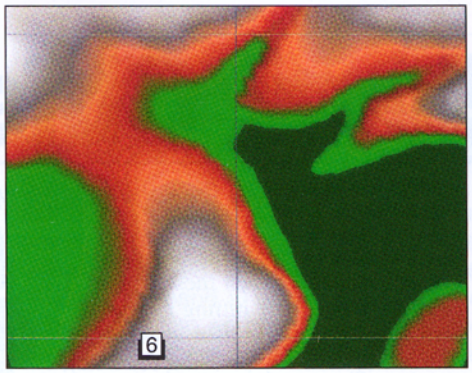
In Figure 2, we see that Kinsale straddles two east-west DEMs. We want to create smaller and more detailed DEMs that can be isolated from the surrounding terrain. Interpolating the two DEMs with 0 Variation (Figure 5), and interpolating the newly created NE and NW corner DEMs again, gives us four small DEMs around Kinsale. Disabling all DEMs except these four, and creating a new planimetric camera, leaves us with the OpenGL view in Figure 6. The width of the camera view corresponds to the width of the four enabled DEMs.

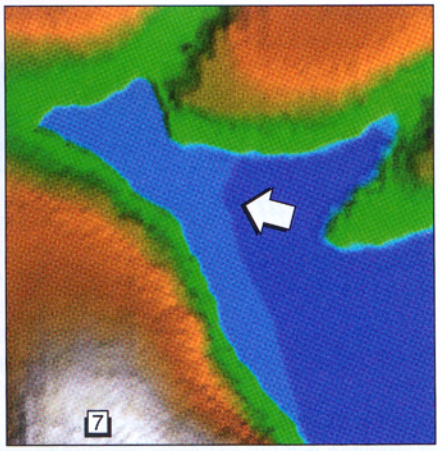
*3D Objects are placed by vectors—a single vertex for stationary obects and multiple vertices for what will be motion paths in LW. When the scene is exported as a .lws scene file, it includes 3D Object vector(s), camera, camera target, light, camera and light paths, and an optional .lwo reference DEM. 3D Studio export includes a .prj scene file with the optional .3ds reference DEM.




Preparing and exporting the terrain in this manner will accomplish two things. First, it will give us a reference terrain surface to model the city on in LW Modeler. More importantly, the default WCS planimetric camera above the exported terrain yields the latitude and longitude of the center of the exported terrain object. Once the terrain object is centered on the X- and Z-axes in Modeler, all objects created in Modeler will be relative to this origin. Back in WCS, we'll create 3D Object vectors at the same latitude and longitude as the exported terrain center. Imported LW objects placed at these vectors will appear at their correct positions relative to the terrain and each other.

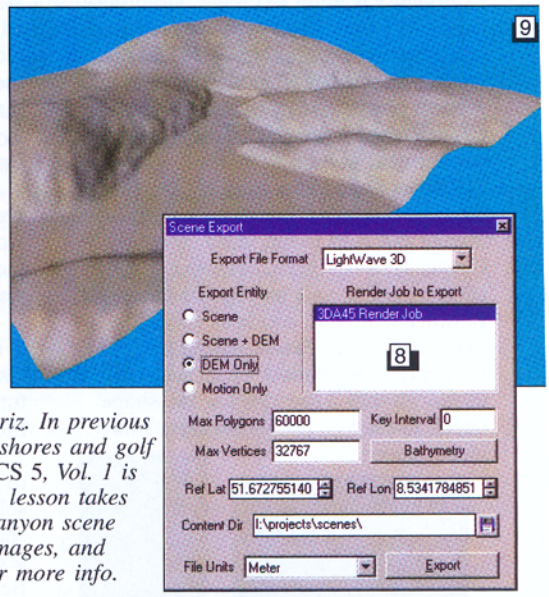
There's another problem we need to head off before exporting the terrain as a LW object. As often happens with terrain data covering large areas, the terrain at modeling scale is too irregular to build our LW town on. Integrating 3D Object houses, roads, and walls with the existing WCS terrain could lead to problems in closer views. Excavating the town site with an Area Terraffector removes the terrain where our town will be. Figure 7 shows the resulting terrain with an elevation driven color gradient Ground Effect with Lakes disabled. We will fill in the hole we just dug (*arrow*) with LW terrain when we build the city.




## From WCS terrain to LW object

To export the WCS terrain and Area Terraffector as a single LW object, we render a default planimetric view as an elevation file (see **www.cherba.com/resource** for details) and import it back into the project. The scene is exported as DEM only, selecting the elevation file just rendered (Figure 8). With the terrain object opened in Modeler, we center it along the X- and Z-axes, and we're ready to start building our town (Figure 9), which will have to wait for another article.




***R Scott Cherba*** *is a photographer, animator, and WCS trainer in Tucson, Ariz. In previous issues he has written about creating seashores and golf greens in WCS. His* Getting to Know WCS 5, *Vol. 1 is available on CD and video. The 90-min. lesson takes you through a re-creation of a Grand Canyon scene with realworld terrain data, reference images, and WCS Components. Visit* **www.wcs5.com** *for more info.*

**Framing a Story** *Continued from page 25*

impression that actors are in the middle of a fire or an explosion when in fact they are at a safe distance. Ridley Scott did that in *Gladiator* when he positioned his lead actor seemingly close to the tigers in the Coliseum during the shoot.

Realistic DOF is mostly overlooked by 3D artists. But, without controling depth of field, you are depriving yourself of a powerful artistic and expressive tool, not to mention the fact that a carefully adjusted depth of field will add much realism to your images.

If your 3D application doesn't have DOF blur as a built-in or add-on feature, you can produce the effect with Z-buffer rendering. A Z-buffer render is a black and white image (Figure 3B) where variations of gray indicate the distance of objects from the camera. You can use this render like an alpha layer to drive a blur filter using Photoshop or compositing software.

A Gaussian blur alone doesn't really do the trick (Figure 3C), however. If you are going to go for the most realistic DOF, you should be aware that a true depth-of-field blur includes an additional optical effect known as "blooming." When you film an unfocused light source or a bright object, a halo effect appears on each bright spot (Figure 3D). This kind of effect can be properly simulated with tools such as KPT Blurr for Photoshop, ProOpticsSuite for 3DS Max, and Defocus Dei for Windows. The last is a standalone to use with any program that can output a Z-buffer file, but also has plug-ins for Max r3+ and LW5.x/6.x.

## Angle & format terminology

The field of view angle is the horizontal angle covered by a camera. This angle corresponds to a specific focal length expressed in millimeters. A 100mm lens covers a narrower field of view than a 50mm. If the format of your image changes, the horizontal angle changes, too. If your format is wider (like a Cinemascope format), the same lens length will cover a wider angle. In the following example, the camera angle value is measured for a 24x36 format for standard 35mm cameras.

**50mm** is very close to human vision. Using this focal length gives you an undistorted view of your subject. It has a neutral point of view.

**35mm** is the upper end of what qualifies as wide angle. It provides a subtle increase in perspective and in the impact of diagonal lines in a picture, but provides virtually no deformation effect. It is interesting to use when you want to give a more graphical or dynamic tone to your image without it being noticed.

**28mm** is where true wide angle begins. With it, deformation is now visible and perspective effects are heightened. It gives a perception of space that is really bigger than the actual space. Unless you are looking for a burlesque or grotesque effect, 28mm and shorter focal lengths are not recommended for shooting close-ups of people.

**24mm** and smaller focal lengths cause deformation effects that become quite obvious, a bit as if space is warping as the camera moves, and a camera move can cause discomfort in viewers.

**75mm** is the lower end of telephoto values. It will bring you closer to your subject without a really noticeable telephoto effect. It is to telephoto what 35mm is to wide angle.

**100mm** is a real telephoto lens. Here is where you start to lose the sense of depth and perspective, and depth of field becomes shallower. A 100mm lens can easily isolate the subject from its background by having the subject in focus and everything else blurred.

**200mm** and larger focal lengths will seem to put everything on the same plane if you don't use a defocus effect. If you simulate depth of field, you can isolate a very small portion of a subject, like an eye on a face seen from side view. This kind of image will look very much non-CGI.

When you begin working on a scene, you should decide early on which focal values you will use. This ensures that the effect or composition you came up with while modeling and staging does not disappear when you set up your cameras. Try to set up your shots very early, using simple shapes and primitives

*Concluded next page, bottom left*

# Rebuilding Kinsale: 1601

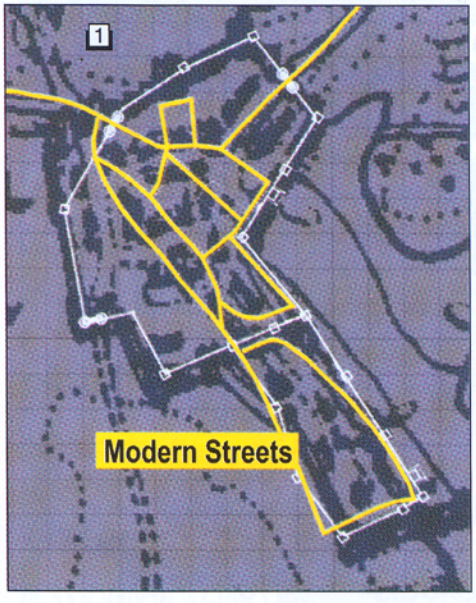
## Part 2: *Historical re-creation using LW5.6 & WCS5*

By R Scott Cherba • Images by the Author

At the end of last issue's article, we had a LightWave object DEM exported from World Construction Set. This is the surface on which we are going to build the old town of Kinsale.

The first thing we'll do is modify the terrain in the area of the old town. This is where a carefully registered backdrop image of the old town and modern day street layout comes in handy (Figure 1). A flat ground level was added to an elevation of three meters within the low area defined by the city walls. Like the WCS DEM object, this is just a reference for modeling in LightWave. We'll build the final ground surface later on in WCS with an Area Terraffector.

Town walls, towers, and gates were modeled from photographs of existing period structures in the UK. As often happens in historical reconstructions, where original structures and documentation no longer exist, the final product




is very much an educated guess based on the best information available (Figure 2). Models were kept as simple as possible. WCS doesn't have any limitations on 3D Object point count, though OpenGL response time will slow as more objects are added to a scene. We won't texture the objects yet, because LW object textures don't import into WCS.

### Matching LW object placement

We now need to correctly place LightWave objects on the WCS terrain. This is a lot easier than you may think, thanks to the way we created and exported our DEM reference. Recall that we used a planimetric camera over the town site to render the elevation file that was exported as a LightWave object. Since the planimetric camera is centered over the WCS DEM and the DEM object was centered in the horizontal plane in LightWave, the LW town center coincides with the planimetric camera location in WCS.

We need a great deal of precision in our lat/lon values, so WCS Preferences/Units: Significant Digits to Display was set to 20.

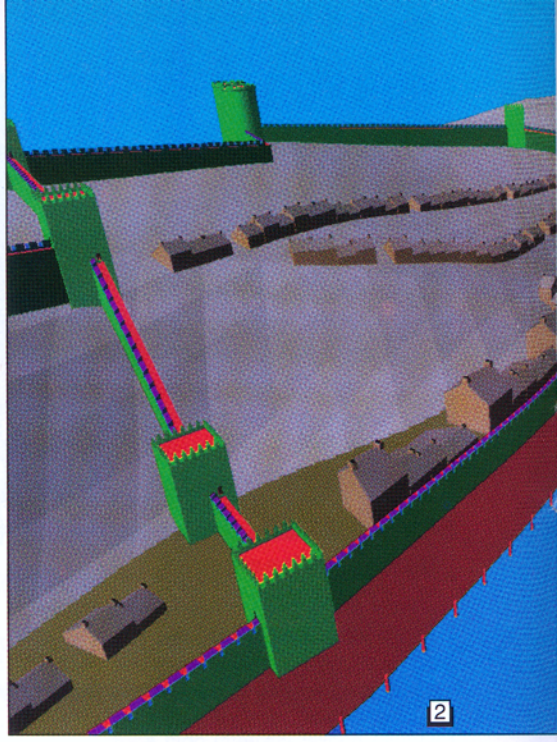
How can we use this information? Create a new 3D Object and vector by selecting its category in the Scene-at-a-Glance and clicking the Create button. Left-click and right-click anywhere in a camera view and name the vector for your first LW object. Expand the 3D Object you just created in the Scene-at-a-Glance and double-click the vector to open the Vector Editor. Open the planimetric Camera Editor to its Position & Orientation page. Copy the Camera Latitude value and paste it into the Vector Editor Y (Lat N-S) box. Copy the Camera Longitude and paste it into the X (Lon W-E) box. Enter 0 for the vector elevation in the Elev box. You now have a 3D Object vector in WCS that coincides with the origin in LightWave.

Use the Vector Editor Create Copy button and create vectors for all of the LW objects. Add new 3D Objects and load the LW objects from file. Expand the Vectors category in the lower Scene-at-a-Glance pane and click, drag, and drop the vectors on their respective 3D Objects.

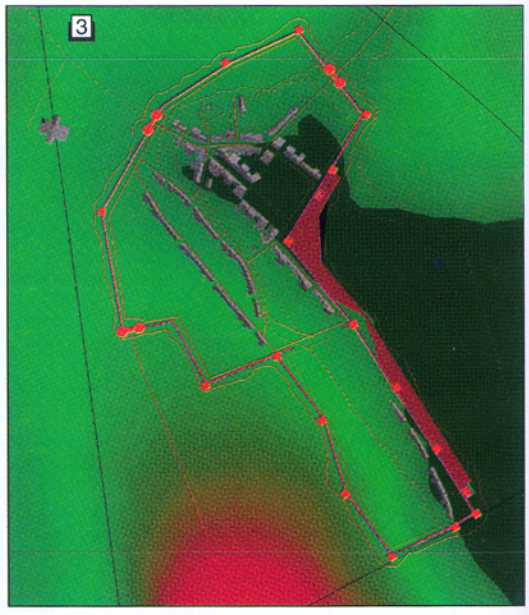
### Just add land & trees

Once 3D Objects become associated with a vector, they show up in the appropriate camera views (Figure 3). If all you see is a box, select Detail Preview in the 3D Object Editor to see a shaded representation of the object.

Since the town was modeled on an exported DEM object, you shouldn't have any buildings or walls floating above or buried below the terrain. Well, not quite true. The city flat was modeled



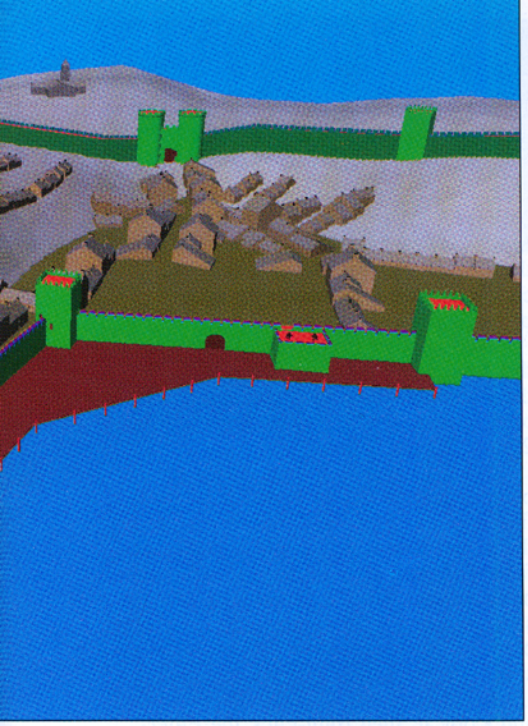

on a 3m elevation reference, so we need to add a 3m absolute elevation Area Terraffector (ATfx). If we select it only to increase elevation and don't use a profile, a vector drawn along the centerline of the town wall will do quite

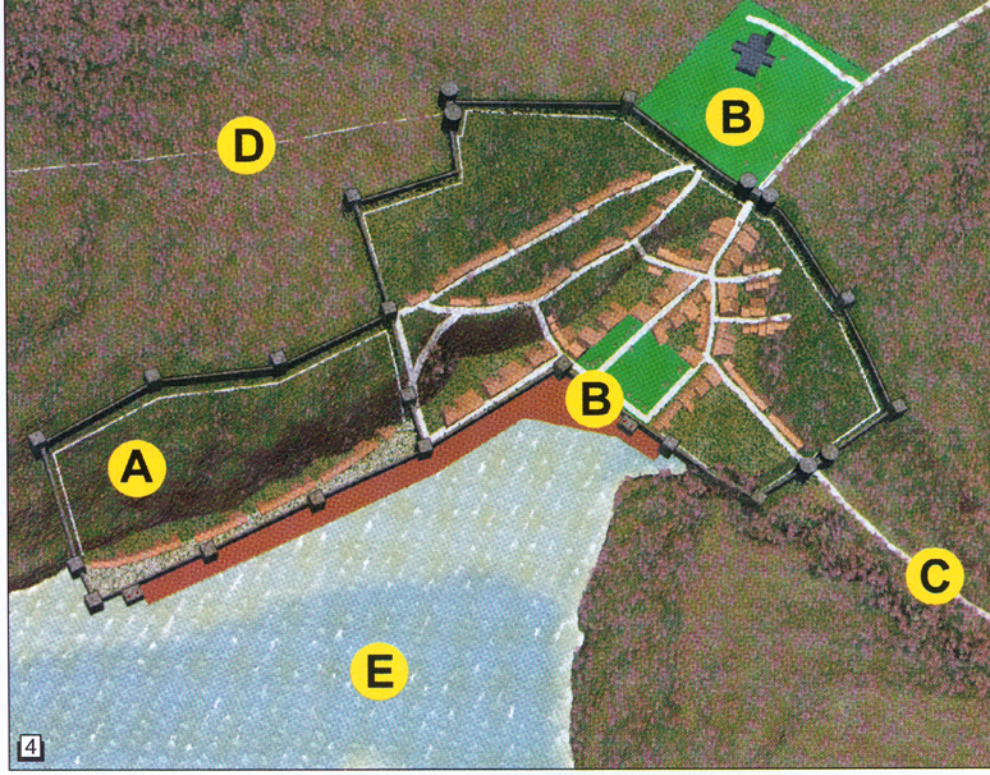



nicely. (Effect Resolution might have to be dropped to 1 meter to prevent a jagged ATfx edge from exiting the town wall along the quay.)

For the Irish winter foliage, a mixture of WCS stock conifers and TreePro deciduous trees and heather-like bushes were used. TreePro made it easy to remove most of the leaves and add fall colors to the remaining few. Trees were rendered as high resolution Targas with alpha channels for easy addition to the WCS Image Library. See Figure 4 for more tidbits of information on what WCS tools were used to complete the town.

*This article also can be applied to Visual*






*Nature Studio (WCS's new sibling for professional GIS users, intro priced at $2,475), where improved features and many functions not available in WCS can increase both realism and working speed.*

***R Scott Cherba** is a photographer and animator in Tucson, Ariz. who has been writing about WCS since* **3DA***#42. Visit him at* **www.cherba.com/resource** *for article supplements, and for WCS/VNS comparison renders related to this article.*

**Fig. 4. (A)** Vector bounded regrowth Ecosystem around town. Global mixed winter forest Ecosystem copied, Rules-of-Nature tweaked, and Ecotype densities varied.

**(B)** Vector bounded park Ecosystems with Foliage Object trees along perimeter. WCS Grass Ecosystem from Content CD with modified texture. Foliage Objects pull trees from global Ecosystem. Churchyard enclosed by Wall 1m Fieldstone Component Terraffector from Content CD.

**(C)** Primary road Terraffectors. Stock WCS Dirt Road Component with shoulder Ecosystem matched to global Ecosystem.

**(D)** Secondary road Terraffectors. Stock WCS Dirt Lane Component with shoulder Ecosystem matched to global Ecosystem.

**(E)** Ocean based on stock Lone Tree Ocean shipped with WCS.