

What's New in Carmenta Engine 5.3

Carmenta Engine 5.3 contains over 100 improvements in total. Here are a few new features that we would especially like to highlight.

For a full list of new features and improvements, refer to the release notes that are included in the installation packages.

Tightly Integrated OGC Map Server Support

With the improved support for retrieving maps from OGC compliant map servers like Carmenta Server, it is now easier than ever to streamline your map management.

Instead of copying map data and map configurations to every machine in your system, the data and configurations are only installed once, onto a map server. This server can be a local, remote or even cloud based server.

Auto-Configuring WMS Connections

Carmenta Engine 5.3 makes it extremely simple to configure map server connections. All you have to do in the new `OgcWmsLayer` is set the URL to the map service – everything else is automatically set up. Another new feature is that from the application's perspective, the layers of a WMS service behave just like local map layers. This means for instance that server layers will show up automatically in the user interface of your application.

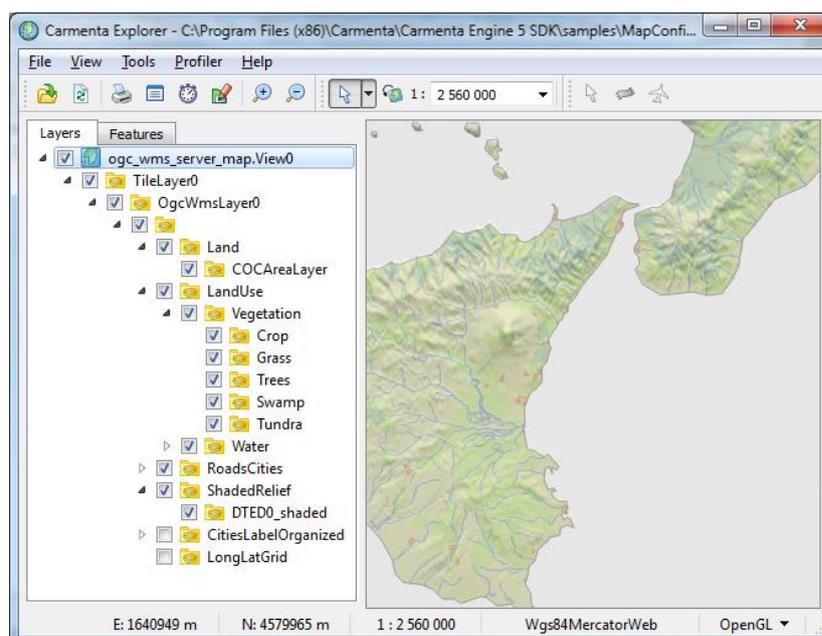


Image 1. Remote OGC WMS layers behave just like local layers. Remote layers show up in layer controls where they can be individually enabled and disabled.

WMTS Support

Another improvement is the support for the new WMTS tile based map service, an OGC standard. The new OgcWmtsLayer is just as quick to set up as the WMS layer, but since it is tile based it is faster and has a more efficient cache.

Improved Background Loading Progress Indication

The new events - MapControl.Busy, MapControl.Idle, View.Busy and View.Idle - make it straightforward for applications to show indicators, such as a turning wheel or a progress bar, when any of the background threads in Carmenta Engine are busy loading or rendering data. This can greatly improve the user experience when map layers are being loaded from slow servers or from slow data formats.

Advanced Map Typography

Fast and Easy Label Deconfliction

A major problem in map making is being able to place labels well, usually with the difficult goal of placing as many labels as possible without overlap. In Carmenta Engine 5.3, the label and symbol organizing functionality has been given a complete makeover. The new, improved LabelOrganizingLayer is now much easier to start using, produces better results, is faster and has several useful new features.

- Quick and straightforward to set up. The default settings work for most maps.
- Labels can have different priorities, which depend on feature attributes. For instance, labels for larger cities are given higher priority than smaller cities, or an aircraft label can be given higher priority than the labels of the waypoints it passes.
- Labels are repositioned or removed to avoid overlap with other labels.
- Duplicate labels can be automatically removed.
- Labels partly outside the map window can be placed where they are fully visible.
- For labels that might have to follow a curved line - usually street names - it is possible to automatically avoid positions where the line is too curved, for a more aesthetic result.
- All label and symbol deconflictions are carried out in real time – no cumbersome preprocessing is necessary.

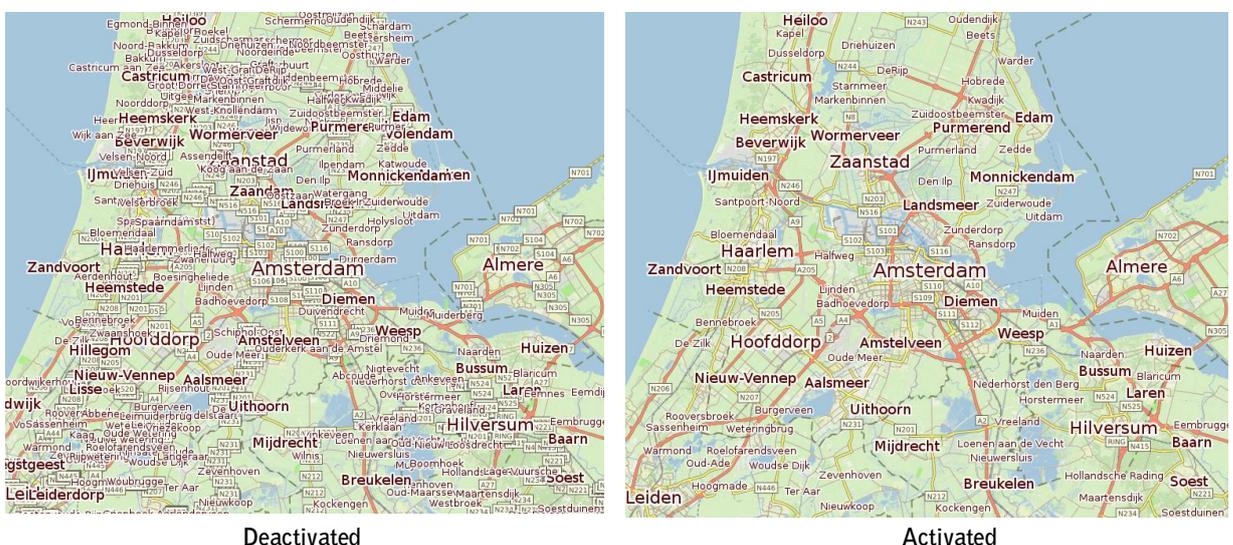


Image 2. City and road number labels with the new label organizer deactivated and activated.

Support for rendering Arabic and other right-to-left languages

Carmenta Engine 5.3 adds support for languages that are written right-to-left. One example is Arabic, which uses a cursive style and where individual glyphs are replaced, depending on the previous or next glyph. Other examples of languages which are now fully supported are Hebrew, Persian, Urdu and Pashtu.

The new implementation, which is of course supported across all Carmenta Engine rendering engines and platforms, also allows mixing left-to-right with right-to-left languages. This function, known as bidirection, is based on the Unicode Bidirectional Algorithm.



Image 3. The new support for rendering right-to-left text.

Productivity Improvements for Map Designers

Improved Carmenta Studio Usability

The Carmenta Studio visual map configuration tool has been improved. Carmenta Studio will now automatically show all possible attributes, names and types of feature that may pass through a layer. The attributes are easily accessible in several drop-down menus, for example when configuring visualization and processing properties and conditions. Previously this required entering the attribute names manually.

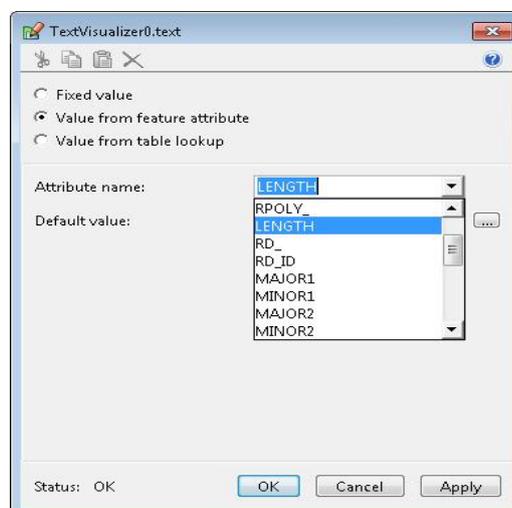


Image 4. Setting the text property to print the value of the attribute LENGTH using the new drop-down menu that automatically displays all available attributes in the layer.

In addition, when configuring properties using lookup tables, the tables can be automatically populated with actual values of the selected attribute.

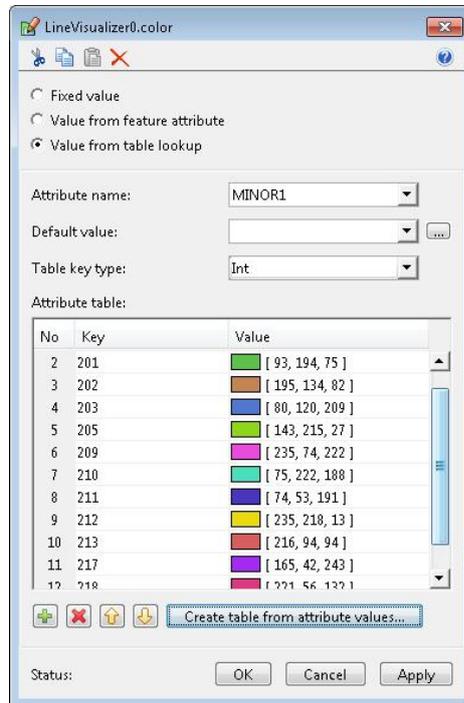


Image 5. The new functionality that automatically fills a colour table with all possible attribute values.

This functionality is also available through the API, which allows developers to add similar functionality to their own applications.

New Documentation Designed Specifically For Map Designers

We have added a new documentation package that is specifically designed for Carmenta Studio users. It shares some of the content with the standard SDK documentation, but elaborates more on Carmenta Studio details, while hiding information that is only relevant for developers.

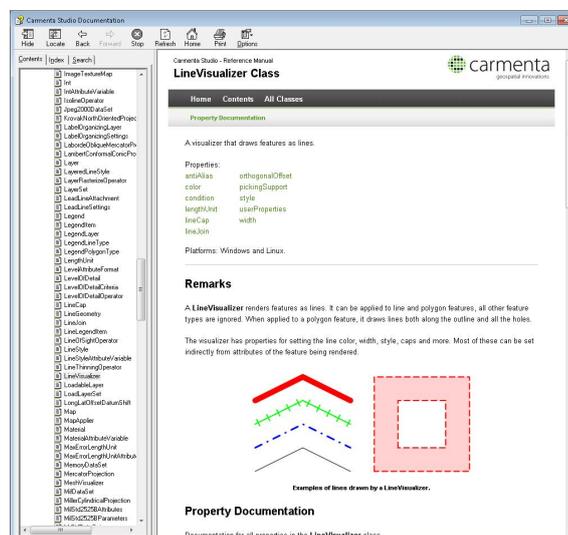


Image 6. The new Carmenta Studio Documentation that is specifically designed for map designers.

New Predefined Symbols

	<code>Symbols.bullseye</code>		<code>Symbols.hollowTriangle</code>
	<code>Symbols.burst</code>		<code>Symbols.nilSymbol</code> (invisible symbol)
	<code>Symbols.circle</code>		<code>Symbols.roundedSquare</code>
	<code>Symbols.cross</code>		<code>Symbols.smallDot</code> (raster symbol)
	<code>Symbols.diamond</code>		<code>Symbols.smallPlus</code> (raster symbol)
	<code>Symbols.hollowCircle</code>		<code>Symbols.square</code>
	<code>Symbols.hollowDiamond</code>		<code>Symbols.star</code>
	<code>Symbols.hollowRoundedSquare</code>		<code>Symbols.triangle</code>
	<code>Symbols.hollowSquare</code>		<code>Symbols.x</code>

New Predefined Line Styles

	<code>LineStyle.highway1</code>		<code>LineStyle.crosses1</code>
	<code>LineStyle.highway2</code>		<code>LineStyle.crosses2</code>
	<code>LineStyle.highway3</code>		<code>LineStyle.dashes1</code>
	<code>LineStyle.marks1</code>		<code>LineStyle.dashes2</code>
	<code>LineStyle.marks2</code>		<code>LineStyle.dashes3</code>
	<code>LineStyle.pluses1</code>		<code>LineStyle.dashes4</code>
	<code>LineStyle.pluses2</code>		<code>LineStyle.dashes5</code>
	<code>LineStyle.railway1</code>		<code>LineStyle.dashes6</code>
	<code>LineStyle.railway2</code>		<code>LineStyle.dashes7</code>
	<code>LineStyle.solid</code>		<code>LineStyle.dashes8</code>
	<code>LineStyle.spikes1</code>		<code>LineStyle.dashes9</code>
	<code>LineStyle.spikes2</code>		<code>LineStyle.diamonds</code>
	<code>LineStyle.spikes3</code>		<code>LineStyle.dots1</code>
	<code>LineStyle.squares</code>		<code>LineStyle.dots2</code>
	<code>LineStyle.triple1</code>		<code>LineStyle.double1</code>
	<code>LineStyle.triple2</code>		<code>LineStyle.double2</code>

New Predefined Patterns

	<code>Patterns.brick</code>		<code>Patterns.stripes1</code>
	<code>Patterns.deciduousTree</code>		<code>Patterns.stripes2</code>
	<code>Patterns.evergreenTree</code>		<code>Patterns.stripes3</code>
	<code>Patterns.dots1</code>		<code>Patterns.stripes4</code>
	<code>Patterns.dots2</code>		<code>Patterns.stripes5</code>
	<code>Patterns.empty</code>		<code>Patterns.stripes6</code>
	<code>Patterns.floodable</code>		<code>Patterns.stripes7</code>
	<code>Patterns.glacier</code>		<code>Patterns.stripes8</code>
	<code>Patterns.grassland</code>		<code>Patterns.stripes9</code>
	<code>Patterns.grid1</code>		<code>Patterns.stripes10</code>
	<code>Patterns.grid2</code>		<code>Patterns.stripes11</code>
	<code>Patterns.mangrove</code>		<code>Patterns.stripes12</code>
	<code>Patterns.mixedTree</code>		<code>Patterns.swamp</code>
	<code>Patterns.orchardPlantation</code>		<code>Patterns.vineyardHops</code>
	<code>Patterns.rice</code>		<code>Patterns.sandAndGravel</code>
	<code>Patterns.sand</code>		<code>Patterns.scrubThicket</code>

Layered Line Styles

Multilayered and multicoloured line styles that previously required multiple layers with separate visualizers can now easily be combined and used as a single line style using the new `LayeredLineStyle` component. Layered line styles are common when visualizing roads. The new line style ensures that intersections between links are displayed correctly.



Image 7. Two multi-coloured road layered line styles displaying an intersection correctly.

Map Performance Improvements

Just as with every release, we have worked hard to ensure that version 5.3 is the fastest Carmenta Engine release yet. This time the team has focused on the following areas:

New, Fully Automatic Cache Management

Carmenta Engine 5.3 has a new way of handling DataSet caches. Rather than specifying cache priorities individually using the cache size property on each DataSet, Carmenta Engine will now automatically ensure that the DataSets that are most frequently accessed get to use more of the cache, while the caches of DataSets that are no longer used (e.g. because they are no longer visible) are automatically released.

Improved DirectX Rendering Performance

Thanks to several low-level changes, the DirectX-based rendering module is significantly faster in Carmenta Engine 5.3. The visual quality of its output has also been improved to the point where it is just as capable as the OpenGL, GDI and Software-based renderers.

This will be particularly useful for applications that use WPF, the Windows Presentation Foundation, since the DirectX support allows Carmenta Engine to integrate fully with these applications. The Carmenta Engine WPF map control has also become more robust in version 5.3: In scenarios where DirectX is not available, it can now switch to a fully software-based rendering mode.

New VPF DataSet

VPF, Vector Product Format, is a geospatial data format used for maps such as VMAP (levels 0, 1 and 2) as well as the World Vector Shoreline (WVS+). Carmenta Engine 5.3 includes a completely rewritten VpfDataSet component.

The new reader is several times faster than the old reader, thanks to the use of a new spatial index. It is also more fault tolerant, allowing it to read virtually any VPF distribution. Another new feature is full support for the VPF-based aeronautical obstacle format, VVOD.

Automatic Point Reduction to Improve Rendering Frame Rate

The TileLayer component can now automatically perform a scale-dependent reduction of the number of points in line and polygon geometries, without sacrificing rendering quality. This makes it much easier to work with, and speeds up the rendering of vector data outside its native scale range.