Release guide

LuciadRIA 2022.1

26 January 2023

Contents

About this release

The 2022.1 release of LuciadRIA focuses on Developer Experience. We updated the controller and editing API, allowing intuitive and tailored map interaction specific to your application. In 3D, you can now offer a richer operational picture thanks to the introduction of draping of vector data over meshes, instancing of 3D icons and bloom effects to attract the user's attention to important elements on the map. This release also offers extended Open Geospatial Consortium (OGC) Symbology Encoding (SE) support for precisely styling vector data, following a given style guide.

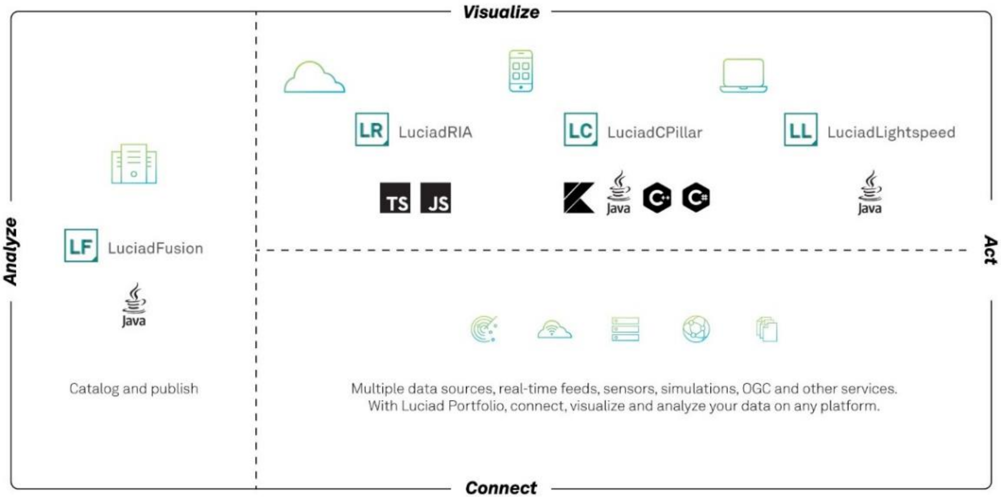

Visualize
LR LuciadRIA
LC LuciadCPillar
LL LuciadLightspeed
TS
JS
Java
Java
Analyze
LF LuciadFusion
Java
Catalog and publish
Act
Multiple data sources, real-time feeds, sensors, simulations, OGC and other services.
With Luciad Portfolio, connect, visualize and analyze your data on any platform.
Connect

Figure 1: The Luciad portfolio

Figure 2: This release brings an even richer operational picture

Benefits of the new features

Intuitive map interaction using updated map controller and editing API

Based on feedback we gathered from our many LuciadRIA developers, we are pleased to announce that for the 2022.1 release, we updated the LuciadRIA map controller and editing API and added extra documentation. These updates cover a set of topics that we list here in arbitrary order.

Interaction with the map

First, we added documentation on the default behavior of the map, both with mouse and touch device input.

Then we improved customizing this default behavior. While it was already possible to add custom controllers, we added an API that exposes the default controllers (e.g., pan, zoom), so you can extend them. Also, there is a new composite controller for chaining controllers. These are only a few examples, but the complete set of changes ensures that you can tailor the map interaction to your project needs. These new API additions will reduce the custom development needed to reach the required behavior.

Add undo/redo support for on-map editing to your application

When users need to do on-map editing of data, undo/redo support is very convenient. To this end, LuciadRIA now offers a new API with the concepts Undoable and UndoManager. The exact behavior of your system will depend on your users' specific needs. Therefore, LuciadRIA doesn't make any assumptions about how undo/redo works in your application. You decide when to create Undoables, what they do and how they're managed. We provide documentation, samples and articles to help you make the right decisions.

Fine-tuned tweaking of feature creation and editing

LuciadRIA introduces a new API to customize creation and editing behavior. The new Editor and EditHandle concepts give fine-tuned access to how features or shapes should be created and edited on the map. Users can implement a custom editor or extend from one of the many out-of-the-box implementations available for LuciadRIA's shape types. The new API enables a wide range of use cases, including:

- Restricting editing operations, such as disabling translation or only allowing to move the individual points of a trajectory
- Synchronizing editing changes between the map and a user interface widget, showing shape properties like the coordinates of an area of interest
- Customizing the look of editing handles or cursors
- Constraining editing such as limiting the width of a buffer (corridor) shape
- Changing the minimum and maximum height of extruded shapes
- Supporting the creation and editing of custom shapes

A new article, "Customizing creation and editing," discusses the possibilities in more detail.

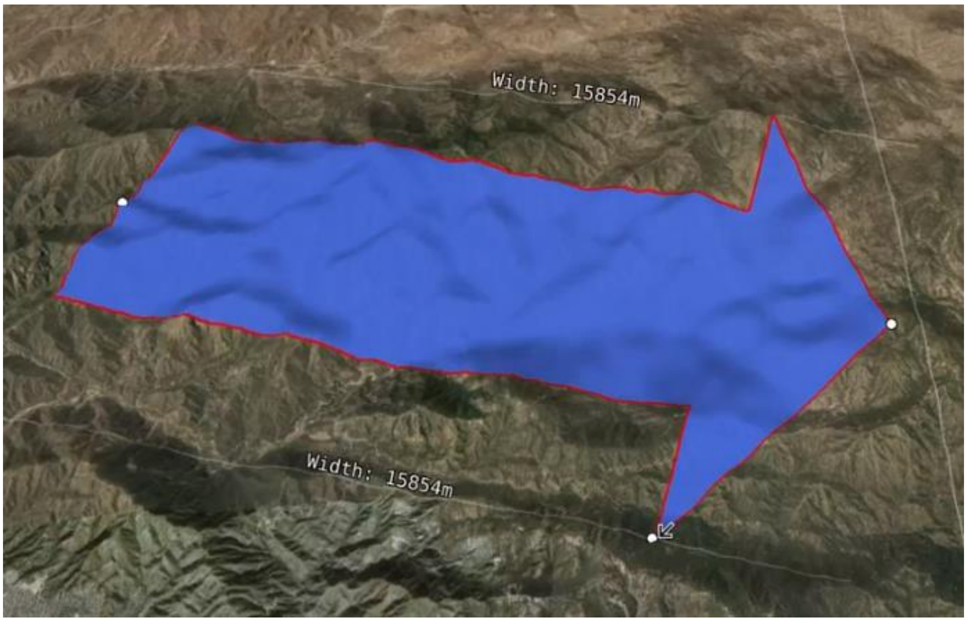


Figure 3: The Create and Edit sample show the creation and editing of a custom arrow shape, including helper lines and labels that show the arrow width

Sample code to get you started

For the new undo/redo capability, several samples show you examples of use cases, such as undo/redo for selection, creation, editing and removal of features. More specifically:

- The module samples/common/util/SampleUndoSupport implements commonly used undo/redo operations in LuciadRIA samples, such as undo/redo for selection, creation, deletion and editing of features
- The “Create and Edit” sample shows you how to wire undo/redo buttons in the UI to an UndoManager
- The “Geolocate” sample illustrates how to use the undo/redo API with a custom controller implementation

To illustrate the new creation and editing capabilities, we extended the “Create and Edit” sample with a “Custom Editing” theme. This theme shows how to implement use cases, such as disabling translation, disabling editing for selected points of a shape and creating a custom arrow shape with helper lines and constraints.

We created or extended the following articles to explain the new features and how they fit together with existing capabilities:

- “Customizing creation and editing”
- “Managing user input with LuciadRIA controllers”
- “Implementing custom user interaction”
- “Adding undo/redo support to your application”

A rich 3D operational picture

LuciadRIA offers many capabilities to represent a digital twin of either your city, site or theater of operations. Whether you have 3D meshes, point clouds, panoramic imagery, live tracks or more, you can bring them together into one single view. What is next? Let’s go back to use cases. One use case is that of the operational picture (OP). The key element in an OP is bringing relevant information together, combined in such a way that a human operator can come to conclusions by looking at the combined information. With this release, we focus on elements that improve the OP. We selected draping vector data over 3D meshes, instancing of 3D objects and bloom effects. Each of those is detailed in the following subsections.

Combine 2D features with 3D meshes data sets via draping

Once the digital 3D model loads into your 3D view, you can augment it with operational data. Consider 2D vector or raster data. In the recent past, when most map views were 2D, these would be overlaid on each other. Now, we need to combine them with 3D information. Depending on the data set, it makes sense to drape it over the terrain only (like a satellite image) or to drape it over 3D content (like the area of a city that will be affected by the shadow of a sizeable new construction). The LuciadRIA 2022.1 release makes it possible to drape 2D content over 3D mesh data (See Figure 4 and Figure 5).

LuciadRIA not only visualizes 2D content but also allows interactive on-map editing of 2D features draped over 3D meshes. You can select the draped content or hover over it with the mouse (See Figure 45).

LuciadRIA API users can configure 3D tiles layers as "drape targets." All data will then be draped on top of the 3D tiles layer. The default is false. In that case, the data will only be draped on the terrain and stay below the 3D tile data.

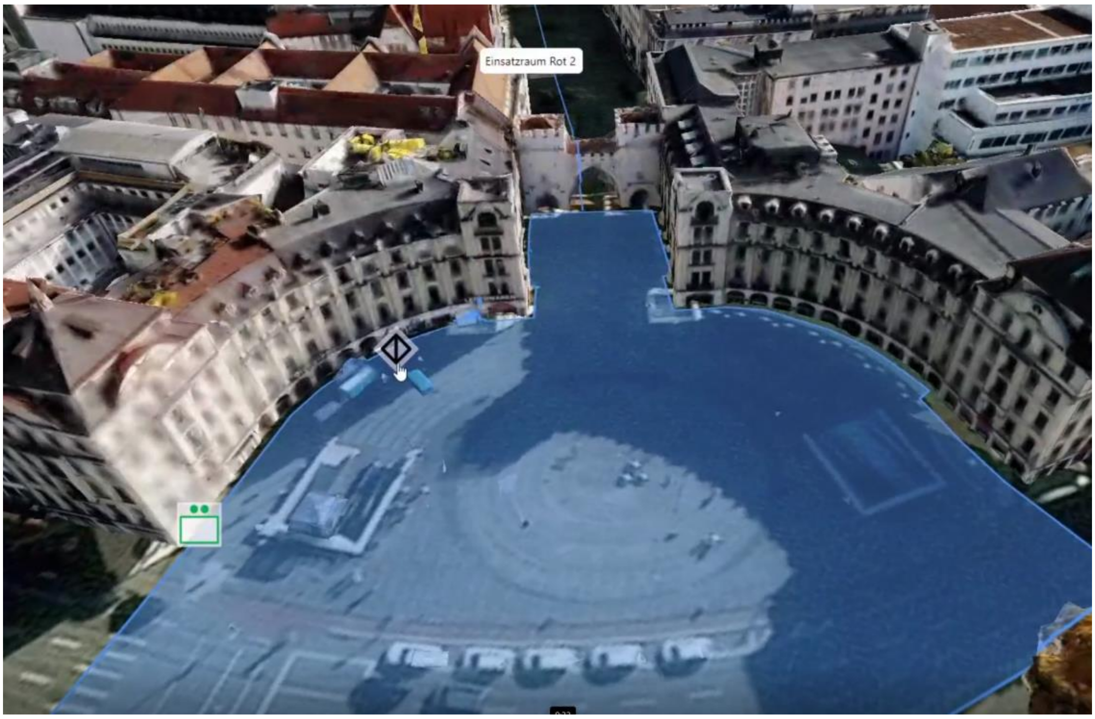
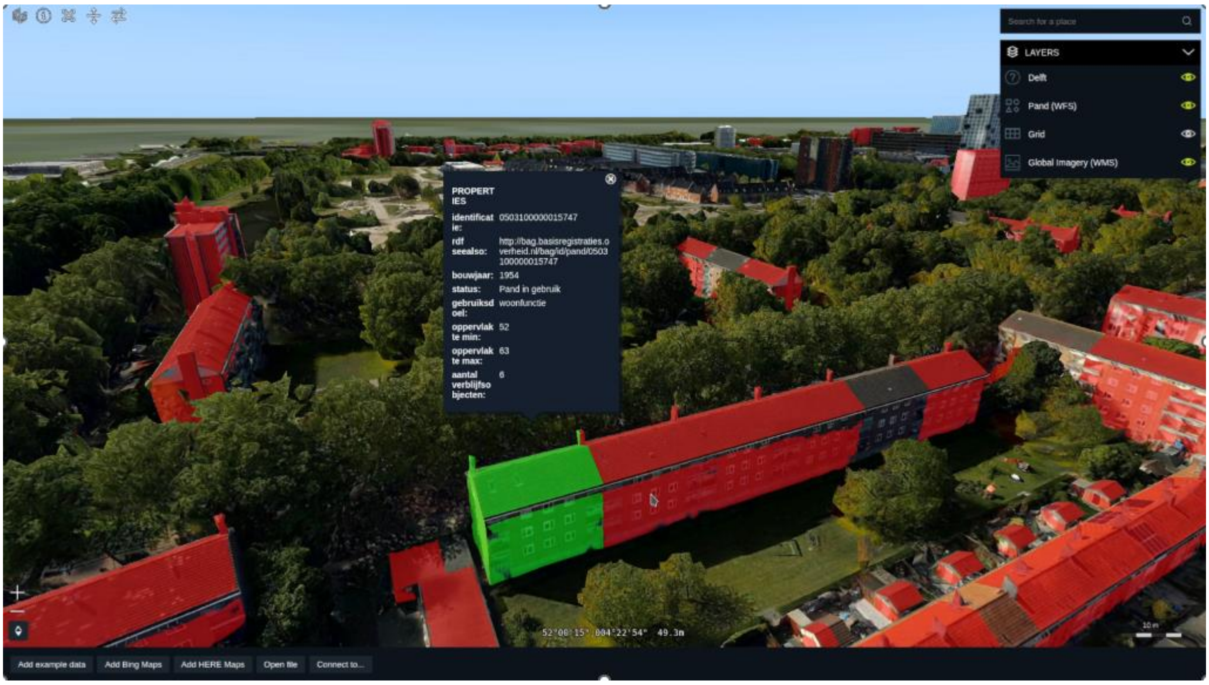


Figure 4: Draping of an area of interest on a 3D city model



PROPERTIES

identificatie: 0503100000015747

rdf seealso: http://bag.basisregistraties.overheid.nl/bag/id/pand/0503100000015747

bouwjaar: 1954

status: Pand in gebruik

gebruiksdoel: woonfunctie

oppervlakte min: 52

oppervlakte max: 63

aantal verblijfsobjecten: 6

Figure 5: Combination of building polygon data and a 3D building model, enabling users to select and access building information

Sample code to get you started

We extended the “OGC 3D Tiles” sample with a walking tour in the center of Marseille draped on top of the Marseille 3D tiles mesh data. The sample code shows how to configure the Marseille 3D tiles layer as drape target and how to drape the walking tour on top of this layer.

Bandwidth-efficient 3D scenes thanks to instancing of 3D icons

3D data sets that originate from scans contain all data elements. If an aerial scan captures a city, it models all houses, and if it captures a forest, all trees are there. The complete data set is encoded in a tile pyramid, for example. But if the data source contains 2D positions that you need to visualize with 3D models, then it makes less sense to explicitly add every 3D model and create a tile pyramid. It is more efficient to have one model per type of 3D icon. For example, if the 2D dataset contains locations of bus stops, and the 3D models are either a 3D model of a bus sign or a 3D model of a bus shelter, another approach is more efficient. In that case, you would opt for instancing: you combine the 2D data set (with bus stops) and the 3D icons (two different ones) at visualization time and position each model as many times as needed to reflect the elements in the 2D dataset. Instancing hence is a memory- and bandwidth-efficient solution.

The 2022.1 release of LuciadRIA adds support for instancing 3D icons in the glTF format or based on custom meshes.

Attract your users’ attention with bloom effects

Bloom is a way to convey intense brightness to the viewer by adding a glow to the bright object. The light then bleeds around these bright objects. Bloom creates the illusion that objects on the map are intensely bright. This effect can add to the perceived realism of a scene. You can also, for example, use it to draw focus upon shapes. The latter is a benefit within a busy common operational picture.

LuciadRIA now offers a BloomEffect and a BloomStyle to add a global or feature-specific bloom effect to your LuciadRIA WebGL map.

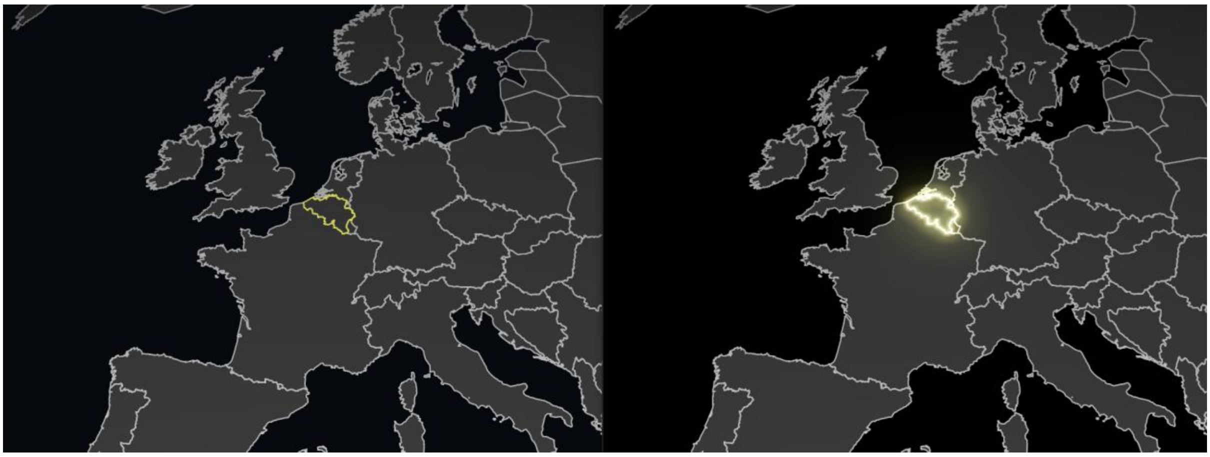


Figure 6: The use of bloom to highlight a feature. The screenshot on the left uses a regular line style; the feature on the right has an additional bloom style.

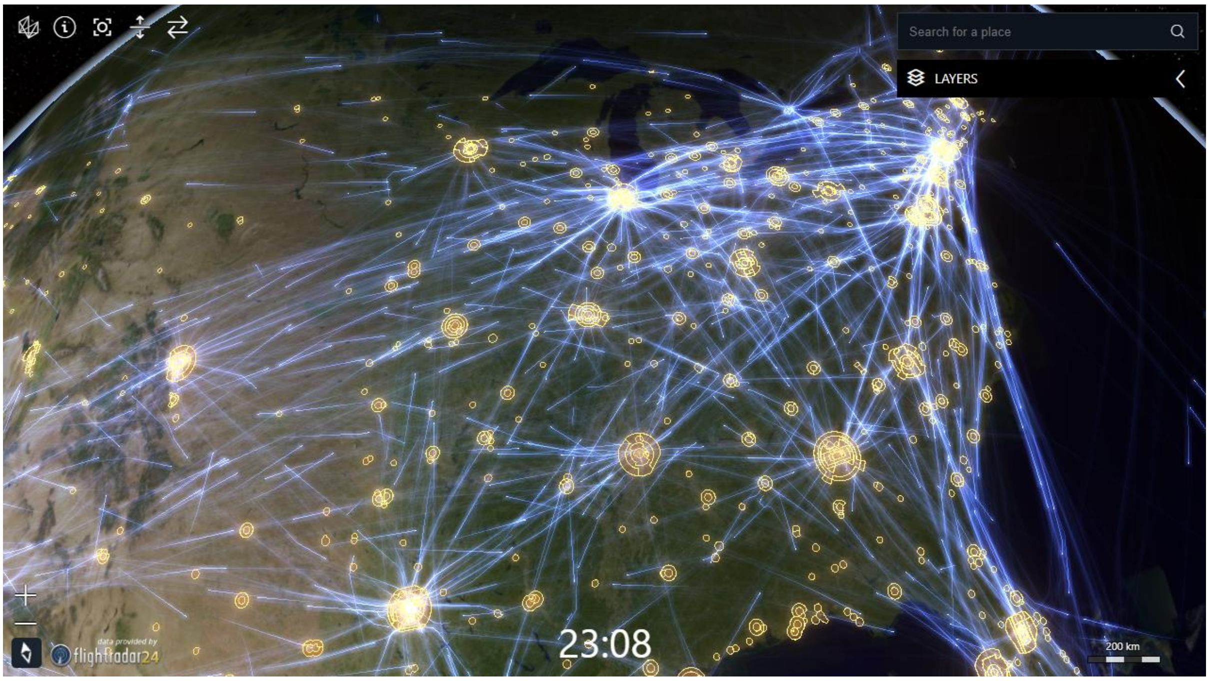


Figure 7: The Trajectories sample uses bloom to highlight airspace data

Sample code to get you started

The following samples show how bloom can be used:

- The “Trajectories” and “Vertical View” samples to highlight airspaces data
- The “Panorama” sample to highlight waypoints when you hover over them in the main map
- The “3D Icons” sample to highlight the billboard on the aircraft carrier

The “Configuring WebGL Map effects” article discusses the use of the bloom API in more detail.

Extended OGC SE styling capabilities

OGC Symbology Encoding (SE), previously called Styled Layer Descriptor (SLD), is a standard for cross-platform and cross-vendor style definition. People widely use OGC SE to style data and to exchange styles between different components in an interoperable way. This 2022.1 release adds the following OGC SE styling capabilities to LuciadRIA:

- To extend the expressive power of OGC SE's label styling options, you can define a label box in a text symbolizer. Aligned with LuciadLightspeed and LuciadFusion, this relies on vendor options to add new capabilities to OGC SE.

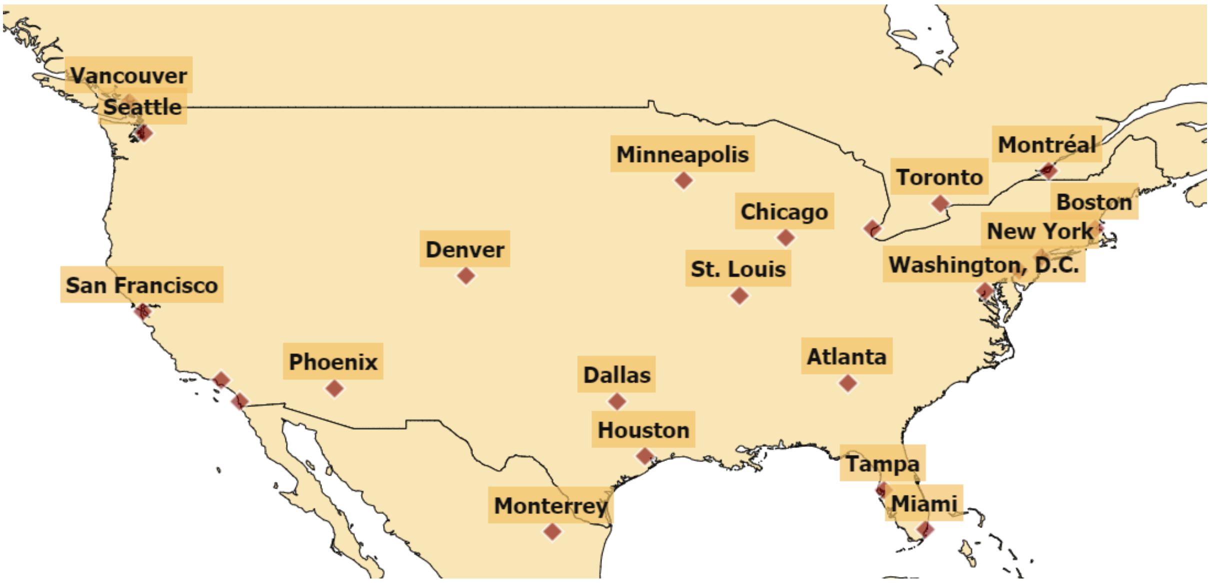


Figure 8: The OGC SE sample adds a label box to the US cities data

- You can use a perpendicular offset on a polygon or line symbolizer. When applied to a polygon, a positive or negative offset value draws the polygon respectively larger or smaller than its actual geometry. LuciadRIA interprets the offset using the unit of measure defined on the symbolizer, which is pixels by default. An example use case of this capability is the styling of class-based airspaces.

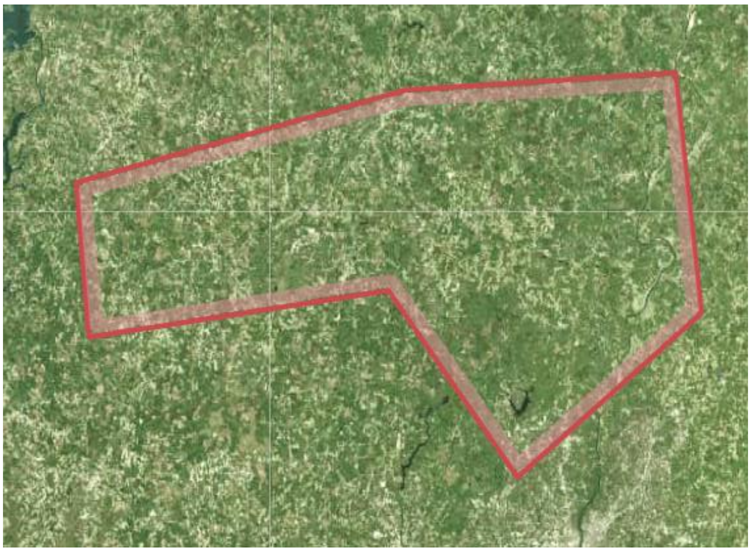


Figure 9: A class-based airspace styled according to ICAO's Annex 4 charting guidelines, using polygon symbolizers with a perpendicular offset

- You can combine a simple (plain or dashed) stroke with icons. The location of these icons can be further adjusted using a displacement or anchor point shift.
- You can use a displacement on a polygon symbolizer to shift the polygon in the X and Y direction. LuciadRIA interprets the displacement value using the unit of measure defined on the symbolizer, which is pixels by default. An example use case is to add a shadow effect to polygons.

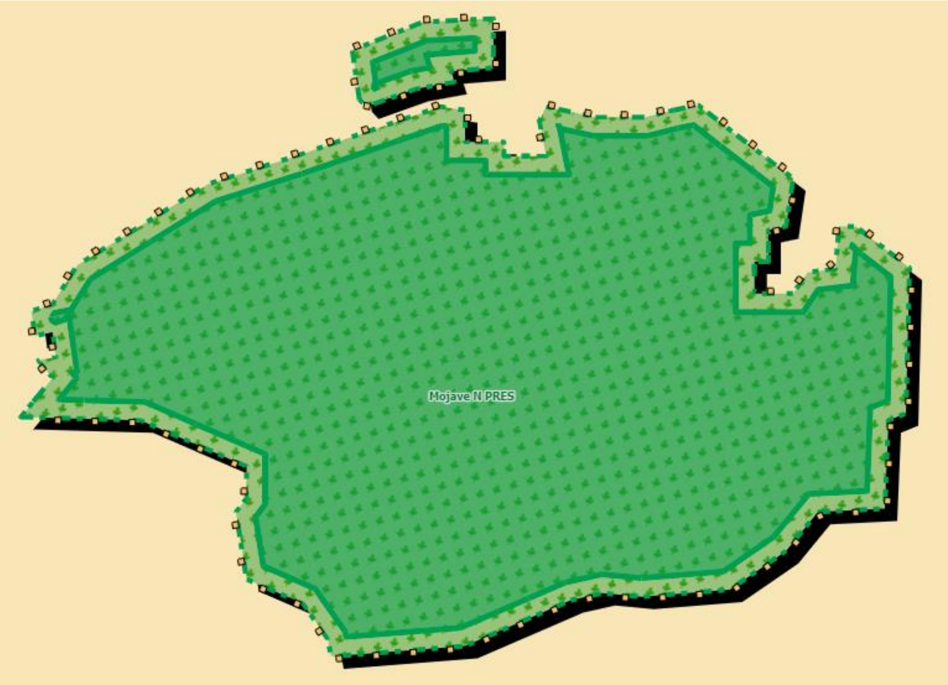


Figure 10: A park area from the OGC SE Sample, combining several of the line and polygon styling improvements. A first polygon symbolizer uses a black fill with a displacement to apply a shadow effect. A second polygon symbolizer defines the park area's fill and outer stroke, which is a combination of a dashed stroke and an icon with a shifted anchor point. A third polygon symbolizer with a negative perpendicular offset is used to draw the smaller green area on top.

- You can now define world-sized graphic fills on a polygon or line symbolizer. Instead of always assuming pixels, LuciadRIA now interprets the size of the graphic used for the fill using the unit of measure defined on the symbolizer.
- Support has been added for the stroke parameters lineJoin and lineCap in WebGL for simple (plain or dashed) strokes. These parameters define how lines should be joined (between line segments) and capped (at the ends of a line). The possible values for lineJoin are "mitre," "round," and "bevel;" the values for lineCap are "butt," "round" and "square."

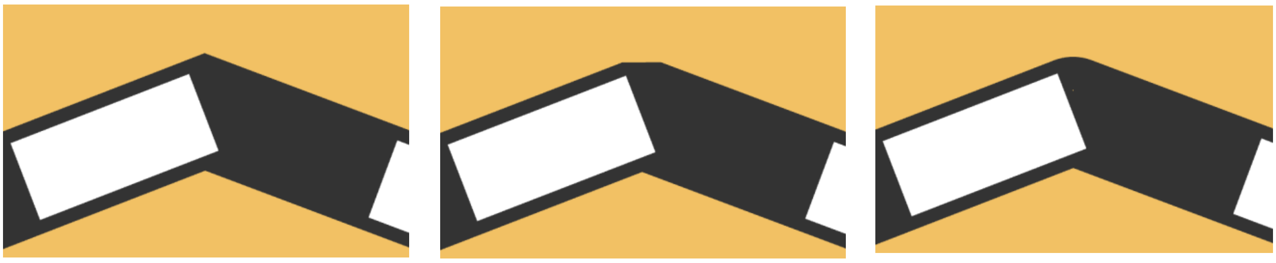


Figure 11: Railroads styled with three different line join settings: left = mitre, middle = bevel, right = round

- The custom icon provider you can plug in to map external graphic references on icon images now also supports SVG.

Sample code to get you started

We updated the styles used by the existing LuciadRIA SE sample to showcase a few of the new possibilities. Because of its interactive style editing panel allowing you to modify styles and see the result in real time, it is also a great starting point to explore the different OGC SE styling capabilities. For further guidance, the new how-to article “SLD styling – Supported versions and capabilities” gives a fine-grained overview of the supported styling capabilities.

Additionally, the new how-to article “How to draw a box around labels” offers more detail on how you can add a label box.

Faster, more flexible and instructive military symbology

LuciadRIA has supported military symbology from its early versions. You can seamlessly integrate tactical graphics and icons for all MIL-STD 2525 and APP-6 standards with a wide range of styling options. The 2022.1 release introduces several new benefits when using military symbology.

Easy look up of symbols and how they are created

The Military Symbols Overview sample provides an overview of all supported military symbols for the MS2525 and APP-6 standards. The 2022.1 release further improves its user experience and functionality; the sample now serves as an interactive version of the supported standards, enabling you to look up symbols and the required anchor points to create them. The top right of the sample offers a search bar for you to enter a symbol name or code. The map fits on the resulting symbol and visualizes its anchor points. When you edit a symbol on the map afterward, for example, by adding points to a line or area, you can easily see where they appeared within the set of anchor points.

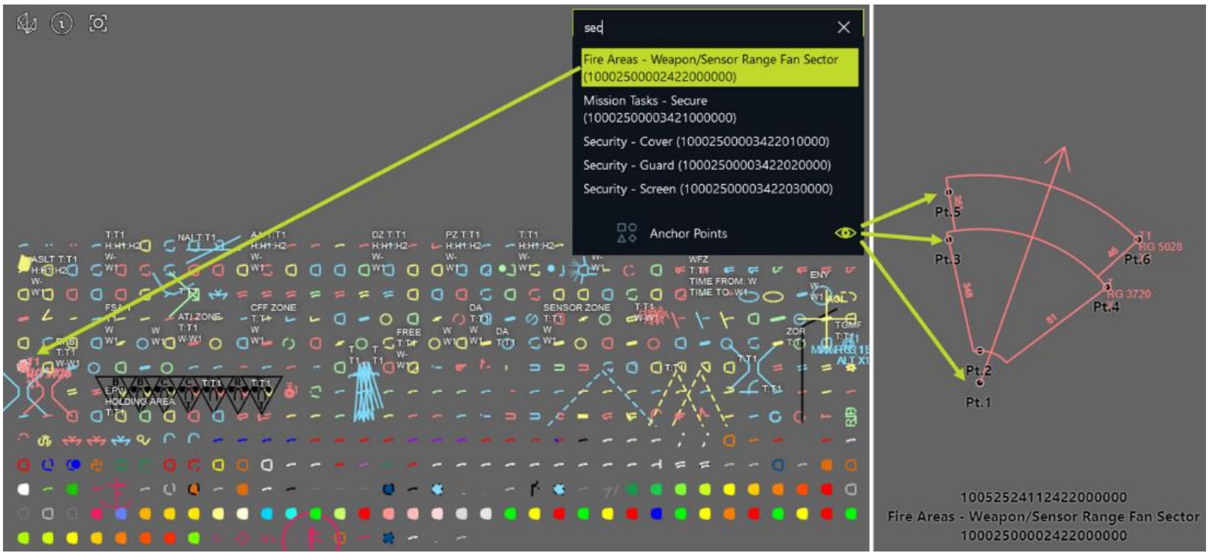

sec
Fire Areas - Weapon/Sensor Range Fan Sector (10002500002422000000)
Mission Tasks - Secure (10002500003421000000)
Security - Cover (10002500003422010000)
Security - Guard (10002500003422020000)
Security - Screen (10002500003422030000)
Anchor Points
Pt.5
Pt.3
Pt.6
Pt.4
Pt.2
Pt.1
RG 5028
RG 3720
10052524112422000000
Fire Areas - Weapon/Sensor Range Fan Sector
10002500002422000000

Figure 12: The improved Military Symbols Overview sample, including a symbol search bar and a layer that visualizes the symbols’ anchor points and the visualization of the Weapon/Sensor Range Fan Sector symbol, consisting of six anchor points

Faster loading of tactical graphics on a WebGL map

Loading many different tactical graphics simultaneously on a WebGL map can be a burden due to the time-consuming process of WebGL shader compilation. To improve the user experience for such use cases, you can now initially render symbols using a fast “skeleton” style. This style represents symbols in a simplified way, with simple lines and without decorations. Afterward, you can repaint them one by one using the accurate “body” style, which does include complex strokes and decorations. This approach prevents delays due to shader compilation, ensuring an interactive user experience. A new article, “Boosting tactical graphics performance on a WebGL map,” has been added to discuss this in more detail.

The Military Symbols Overview sample applies this approach to show the tactical graphics for each supported standard. The symbols are preloaded using the skeleton style and afterward repainted one by one using the body style. The layer control panel tracks the loading progress of the fully styled tactical graphics.

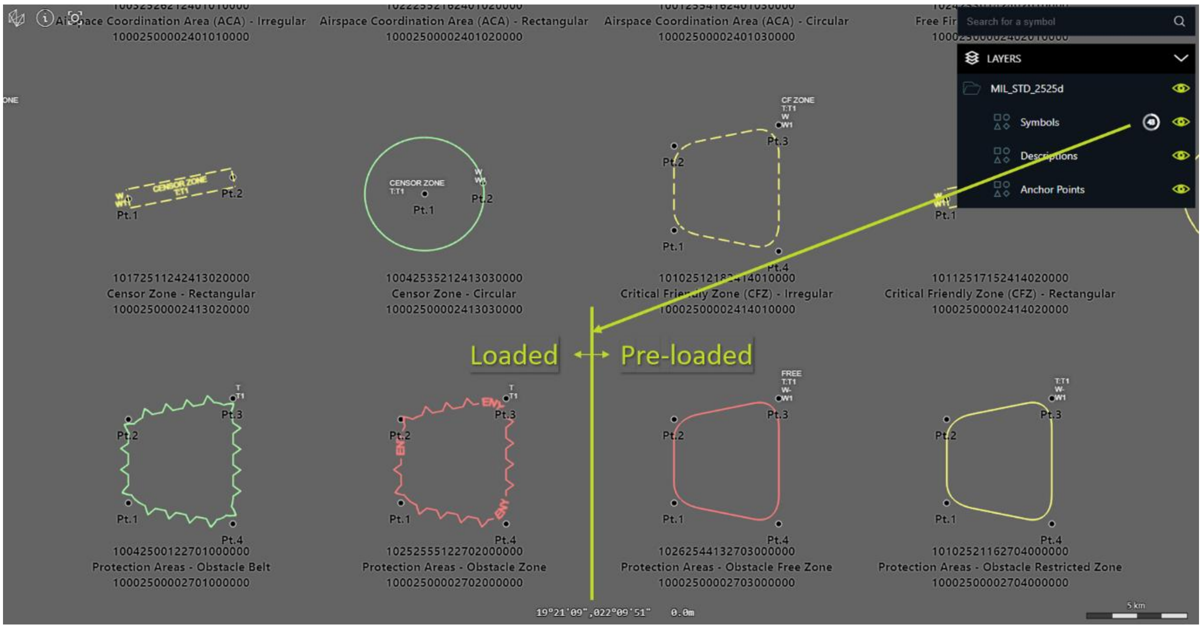


Figure 13: Zoomed in on the progress while loading the tactical graphics symbols in WebGL in the Military Symbols Overview sample. The preloaded symbols rendered using skeleton style are repainted one by one using the body style.

Additional flexibility to style military symbols

The 2022.1 release introduces new styling options to customize the appearance of military symbols on a map:

- Change the style of military symbols based on the paint state: You can customize the style of each military symbol based on whether it renders in regular, selected or hovered mode. The Military Symbols Overview sample uses this capability to style symbols with a halo when you hover over them.

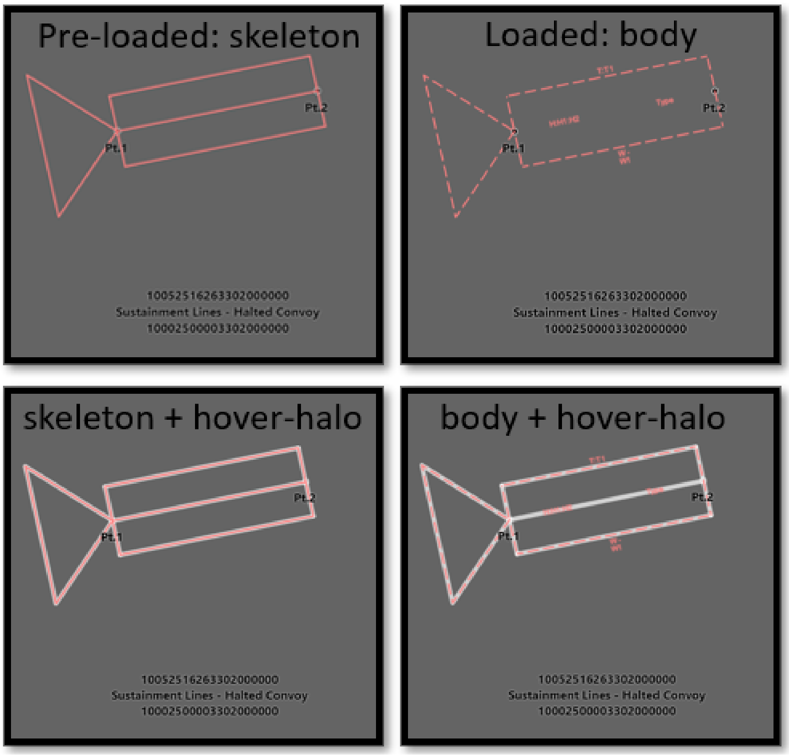


Figure 14: The Military Symbols Overview sample applies a white halo when you hover over a symbol

- Enable or disable label decluttering for military symbols: You can define whether the labels of military symbols on a map can be decluttered, or whether they need to always be visible, regardless of any overlap.
- Customize the length of military symbol speed indicators: You can supply a length ratio to define how a speed value needs to convert to a speed indicator length in pixels.

About Hexagon

Hexagon is a global leader in digital reality solutions, combining sensor, software and autonomous technologies. We are putting data to work to boost efficiency, productivity, quality and safety across industrial, manufacturing, infrastructure, public sector and mobility applications.

Our technologies are shaping production and people-related ecosystems to become increasingly connected and autonomous — ensuring a scalable, sustainable future.

Hexagon's Safety, Infrastructure & Geospatial division improves the resilience and sustainability of the world's critical services and infrastructure. Our solutions turn complex data about people, places and assets into meaningful information and capabilities for better, faster decision-making in public safety, utilities, defense, transportation and government.

Hexagon (Nasdaq Stockholm: HEXA B) has approximately 23,000 employees in 50 countries and net sales of approximately 4.3bn EUR. Learn more at hexagon.com and follow us @HexagonAB.

Copyright

Warning: The product made the subject of this documentation, including the computer program, icons, graphical symbols, file formats, audio-visual displays and documentation (including this documentation) (collectively, the "Subject Product") may be used only as permitted under the applicable software license agreement, and subject to all limitations and terms applicable to use of the Subject Product therein. The Subject Product contains confidential and proprietary information of Intergraph Corporation, a member of the Hexagon Group of companies ("Hexagon"), its affiliates, and/or third parties. As such, the Subject Product is protected by patent, trademark, copyright and/or trade secret law and may not be transferred, assigned, provided, or otherwise made available to any third party in violation of applicable terms and conditions cited further below.

Terms of Use

By installing, copying, downloading, accessing, viewing, or otherwise using the Subject Product, you agree to be bound by the terms of the EULA found here: https://legaldocs.hexagon.com/sig/Licenses/EULA_SA_SIG-Eng_062021.pdf

Disclaimers

Hexagon and its suppliers believe the information in this publication is accurate as of its publication date. Hexagon is not responsible for any error that may appear in this document. The information and the software discussed in this document are subject to change without notice.

Language Translation Disclaimer: The official version of the Documentation is in English. Any translation of this document into a language other than English is not an official version and has been provided for convenience only. Some portions of a translation may have been created using machine translation. Any translation is provided "as is." Any discrepancies or differences occurring in a translation versus the official English version are not binding and have no legal effect for compliance or enforcement purposes. Hexagon disclaims any and all warranties, whether express or implied, as to the accuracy of any translation.

Reasonable efforts have been made to provide an accurate translation; however, no translation, whether automated or provided by human translators is perfect. If any questions arise related to the accuracy of the information contained in a translated version of Documentation, please refer to its official English version. Additionally, some text, graphics, PDF documents, and/or other accompanying material may not have been translated.

Links To Third Party Websites

This Document may provide links to third party websites for your convenience and information. Third party websites will be governed by their own terms and conditions. Hexagon does not endorse companies or products to which it links.

Third party websites are owned and operated by independent parties over which Hexagon has no control. Hexagon shall not have any liability resulting from your use of the third party website. Any link you make to or from the third party website will be at your own risk and any information you share with the third party website will be subject to the terms of the third party website, including those relating to confidentiality, data privacy, and security.

Hexagon provides access to Hexagon international data and, therefore, may contain references or cross references to Hexagon products, programs and services that are not announced in your country. These references do not imply that Hexagon intends to announce such products, programs or services in your country.

Revisions

Hexagon reserves the right to revise these Terms at any time. You are responsible for regularly reviewing these Terms. Your continued use of this Document after the effective date of such changes constitutes your acceptance of and agreement to such changes.

Questions

Contact us with any questions regarding these Terms.