Release guide

LuciadRIA 2024.0

28 June 2024

Contents

About this release

The 2024.0 release of LuciadRIA adds new formats and offers new tools for improving point cloud visual quality. This release also marks the end of the 2D non-WebGL map introduced in 2012. Moving forward, we will focus on WebGLMap.
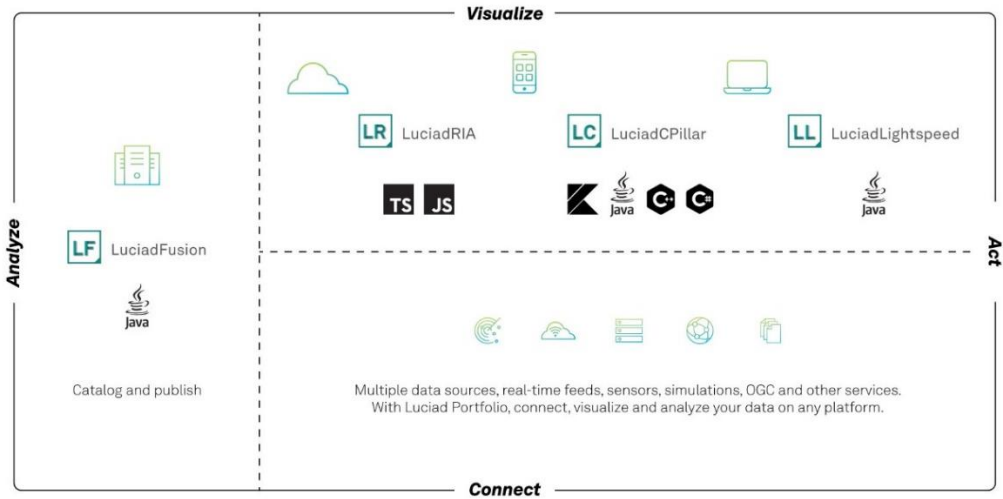

Figure 1: The Luciad portfolio

2D non-WebGL map end-of-life announcement

LuciadRIA was released in 2012 with a 2D map based on an HTML5 Canvas. In 2016, the WebGL-powered hybrid 2D/3D map was introduced, leveraging WebGL support in most browsers. Moving forward, we will focus solely on a WebGLMap. This offers all the capabilities of a non-WebGL map and brings much more to the table.

Starting from LuciadRIA 2024.0, the non-WebGL map will be deprecated. The 2024.0 and 2024.1 releases will be the last ones supporting it. We understand that your applications in the field have a significant lifespan. LuciadRIA 2024 will be actively maintained until the release of version 2027. Contact us at product.management.luciad.gsp@hexagon.com if you plan to extend your maintenance on LuciadRIA 2024 beyond 2027 so we are aware of your project.

In our next major release, LuciadRIA 2025.0, we plan to remove the non-WebGL map. All maps will switch to hardware accelerated. We're committed to making the transition as smooth as possible. Switching to WebGLMap with LuciadRIA 2024 will be as effortless as changing the current constructor call for the map. See
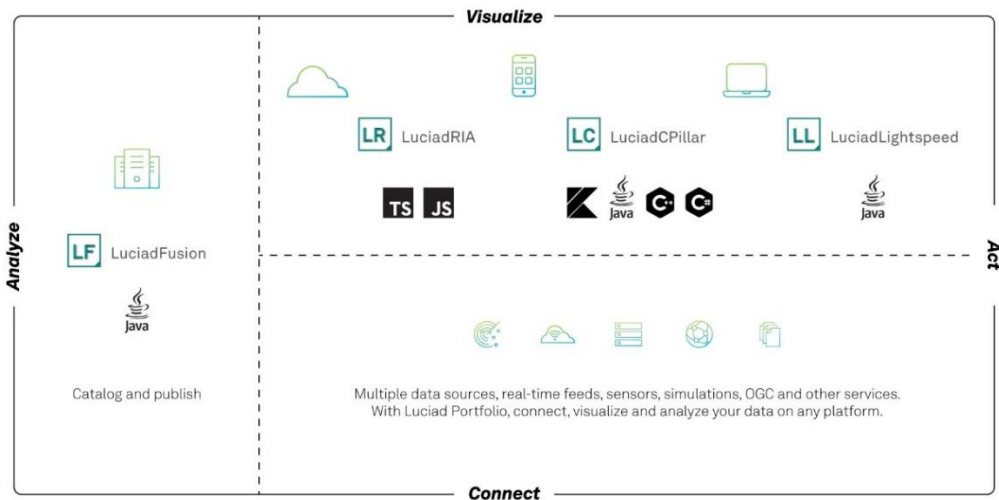

Figure *1*.

The article "Non-WebGL map end-of-life announcement" contains full details to ensure you're fully informed and prepared for this transition. Should you have any questions, please contact us.

Switching to WebGLMap in your application

You can switch to `WebGLMap` by changing the constructor call of your `Map`. Replace

```
1   import {Map} from "@luciad/ria/view/Map.js";
2
3   const map = new Map(domNode, options);
```

with

```
1   import {WebGLMap} from "@luciad/ria/view/WebGLMap.js";
2
3   const map = new WebGLMap(domNode, options);
```

The `WebGLMap` API is a superset of the `Map` API, so no further changes are necessary.

See LuciadRIA benefits for the features available to you when you use `WebGLMap`.

You can find the WebGL system requirements here.

> ℹ️ We plan to keep the `Map` type in the API in 2025.0, as an alias for `WebGLMap`. That way, you won't have to update all usages of the `Map` type to `WebGLMap`, as an argument type in method signatures for example.

Figure 2: A section from the article "[Non-WebGL map end-of-life announcement](#)"

Benefits of new features

OGC 3D Tiles 1.1 connector

We have supported OGC 3D Tiles since version 2018, starting with support for version 1.0. In the meantime, the new version of 3D Tiles has been adopted as an OGC Community Standard[1]. Previously referred to as "3D Tiles Next," version 1.1 of the 3D Tiles Community Standard is designed for streaming high-resolution, semantically rich 3D geospatial data to the metaverse. OGC 3D Tiles 1.1 promotes several 3D Tiles 1.0 extensions to "core" and introduces new glTF extensions for fine-grained metadata storage.

With LuciadRIA 2024, you can now visualize OGC 3D Tiles 1.1.

The specification of 3D Tiles 1.1 is rich. We have not yet received test data for each new feature, so the currently supported functionality includes:

- Implicit tiling
- GLB/glTF as payload
- Support for a number of GLTF 2.0 extensions

Note: glTF data with the aforementioned extensions can also be loaded as 3D icons.

We will continue to complete our support for OGC 3D Tiles 1.1 in upcoming releases, driven by data that we notice and that is reported by you. For example, metadata is not supported, because we have not yet encountered sufficient OGC 3D Tiles 1.1 test data including metadata.

On a similar note: The OGC3DTilesModelDescriptor has two new properties: hasPointCloud and hasMesh. These new properties allow you to conveniently check if your 3DTiles dataset contains a mesh, a point cloud or both.

Sample code/documentation to get you started
The documentation topic "Data formats: OGC 3D Tiles" and the reference documentation for OGC3DTilesModel have been updated to cover the supported OGC 3D Tiles 1.1 functionality.

The "Data Formats" sample and the "OGC 3D Tiles" sample both support OGC 3D Tiles 1.1. See Figure 3.

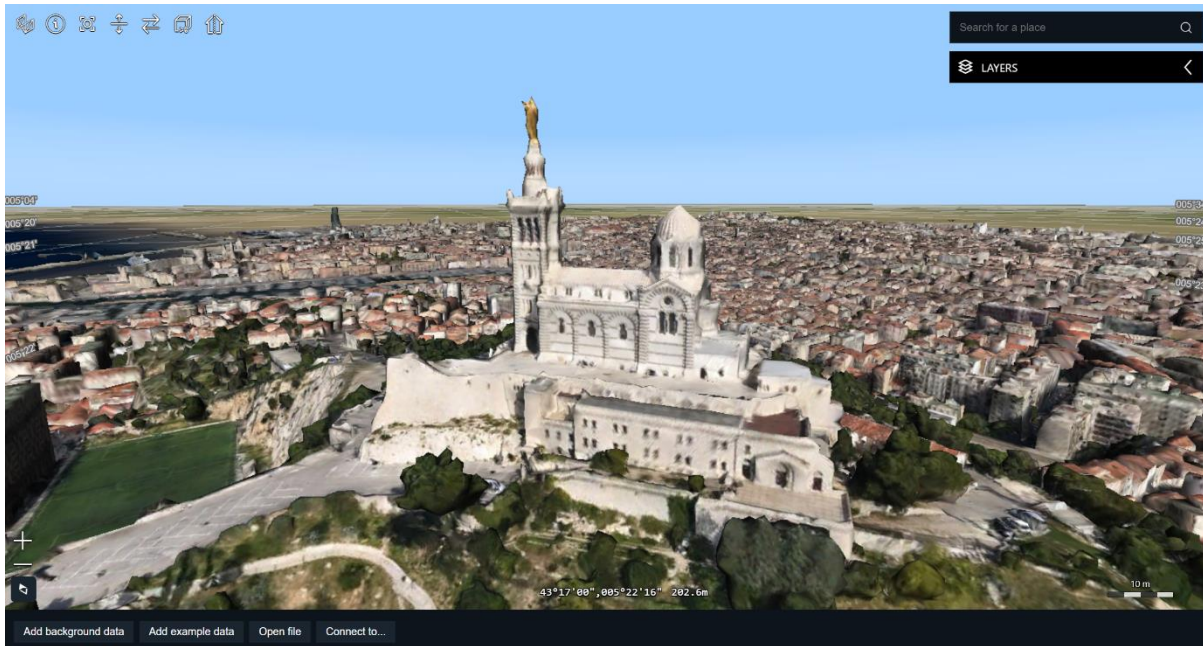[1] www.ogc.org/press-release/ogc-adopts-3d-tiles-v1-1-as-community-standard

Figure 3: The sample "Data Formats" has been updated.

Google 2D Map Tiles connector

As of version 2024, LuciadRIA includes a connector for Google 2D Map Tiles[2]. Google 2D Map Tiles divide the world into an indexed grid. It lets you access and utilize map data efficiently and dynamically at multiple cartographic scales. The Map Tiles API gives you access to multiple thematic geo-datasets, including Google-curated roadmap image tiles and satellite imagery.

The new class GoogleMapsTileSetModel allows you to create a model for the desired Google Maps data, which can then be added to a LuciadRIA map using a RasterTileSetLayer.

Sample code/documentation to get you started
The article "Visualizing Google 2D Tiles" has been added to the documentation to guide you. Basic usage is illustrated in the LuciadRIA Data Formats sample.

---

[2] https://developers.google.com/maps/documentation/tile/2d-tiles-overview
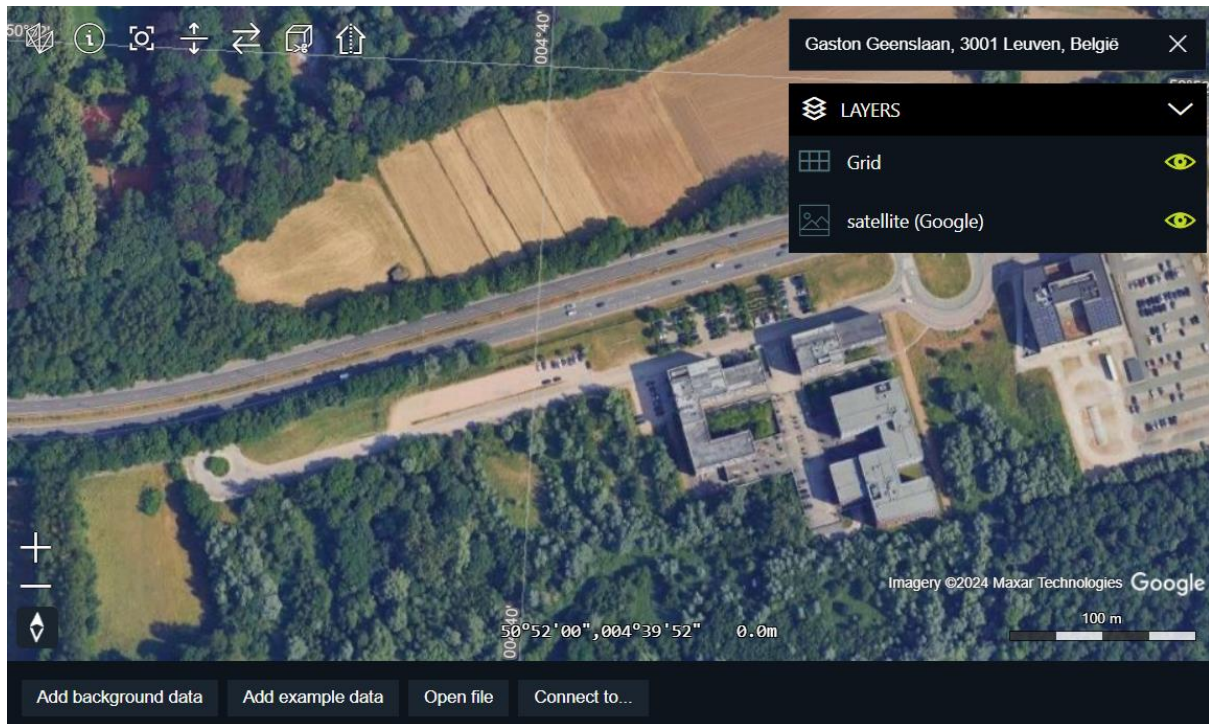
Figure 4: A LuciadRIA view of the Leuven Hexagon office in Google 2D Tiles

Point cloud rendering improvements

We continuously strive to improve the quality of point cloud rendering within LuciadRIA. Applying the right settings can make a significant difference for your application user, as is depicted in Figure 5.

This release brings two additional configuration options for quality and a new feature: point cloud density painting.

Additional configuration options

You can now display point clouds using normal-oriented points. The point orientation setting is enabled by default if the dataset has normal vector values. This setting is illustrated in Figure 6.

In addition, there is a new post-processing filter available for point clouds: blending. When enabled, it merges overlapping points together to increase visual quality. You can see the effect in Figure 7.

Point cloud density painting

WebGL maps support hardware-accelerated density maps, often called heat maps. A density map counts overlapping points on each pixel and colors the pixels according to the amount of overlap.

PointCloudStyle now contains a density property. It allows you to set a ColorMap on the point cloud style. The ColorMap is used to render the point cloud model as a heat map of the point density.

Figure 5: The picture on the left shows an unclear, sparse point cloud. The picture on the right shows this point cloud displayed with high visual quality.

Figure 6: The picture on the left shows a point cloud visualized without considering the normal orientation. The picture on the right shows the result with normal orientation.

Figure 7: The picture on the left shows a point cloud without blending. The picture on the right shows a point cloud with blending enabled, which improves the readability of the text.

Sample code/documentation to get you started
The information about normal orientation and blending has been integrated into the article
"Tuning performance and visual quality of point clouds." For more information on density
painting, see the new "Density painting of point cloud data" article.
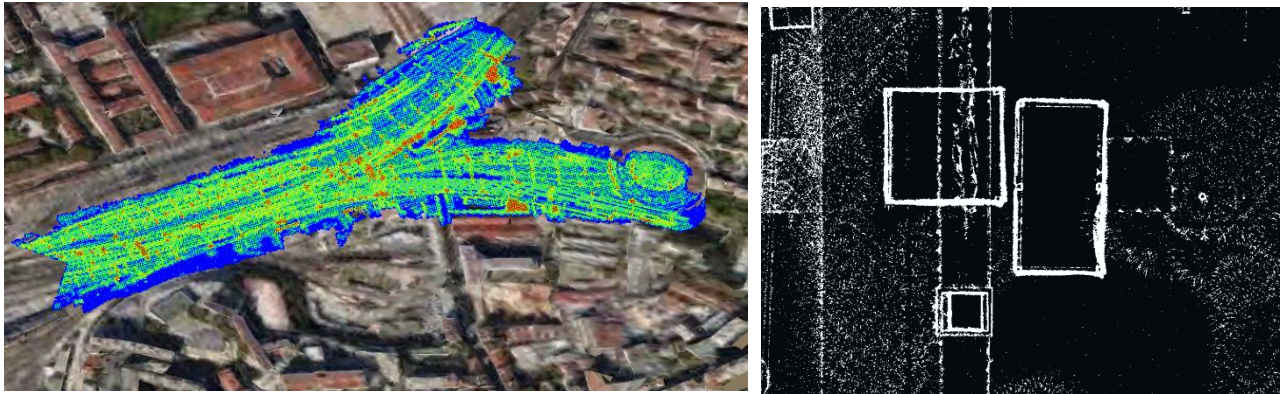
Figure 8: An illustration of point cloud density painting with applied color map (left) or for the specific use case of generating a floor plan (right)

Explore LuciadRIA Toolbox

The LuciadRIA product release installation contains sample code in addition to the main
product. This is the case for all products in the Luciad portfolio. See
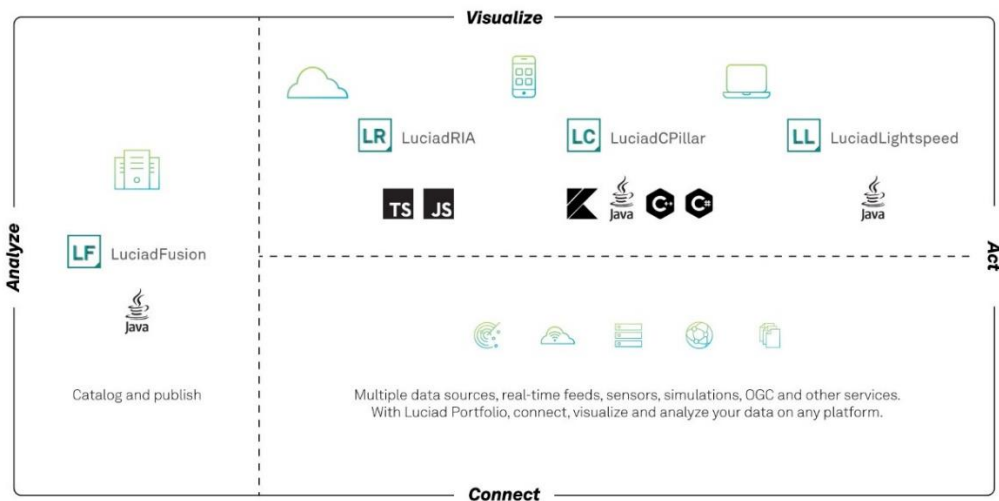
Figure 1.

The purpose of sample code is illustrating how to leverage our API to implement all kinds of
use cases. Over time, the LuciadRIA samples were organized such that they contain
samples/common, a directory with UI components that are used by multiple samples.

In this release, we go a step further and introduce the LuciadRIA Toolbox. This is a
collection of tools and utility code required by many customers. A basic integration API gives

you easy access to the sophisticated features in this toolbox. It is either intended to be used as-is, or completely modified from source code.

Note: Unlike the common sample directory, Toolbox samples don't contain UI, only functionality.

Sample code/documentation to get you started
The guide "Reuse sample code in your own application" summarizes all available sample code and offers best practices for integration into your project.

Other improvements

- It is now **easier to change the behavior of the default map controller**. More specifically, Map now exposes a DefaultController field, which makes it easier to use custom navigation, hover or select controllers on the map. There are also new DefaultController and NavigateController classes that expose the default map behavior. See the article "Managing user input with LuciadRIA controllers" for an update.
- **LuciadRIA now uses WASM SIMD**[3] instructions in some places to improve performance. **Note**: This feature requires browser support for SIMD. Such support was added in Firefox in version 89, in Edge in version 91, in Chrome in version 91 and in Safari in version 16.4.
- **Support for sparse WMTS raster layers**, i.e., WMTS raster layers with a tile-set structure characterized by missing tiles, along the tile pyramid, has been improved. LuciadRIA now allows tiles of higher zoom levels to be displayed even if parent tiles of lower zoom levels are unavailable. This improvement was already released as part of one of the 2023.1 patches.

---

[3] https://github.com/webassembly/simd

About Hexagon

Hexagon is the 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 24,500 employees in 50 countries and net sales of approximately 5.4bn EUR. Learn more at hexagon.com and follow us @HexagonAB.

Copyright

Terms of use

Disclaimers

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.