LuciadCPillar 2026.0
Release Guide

Contents

About this release

The 2026.0 release of LuciadCPillar marks an important milestone for the product. LuciadCPillar rendering now relies on WebGPU instead of OpenGL. While this switch has only minimal impact on the API user, it brings many benefits for the future. Furthermore, this release brings a number of new features such as expression-based line and area styling and filtering, ECW and JPEG2000 support, and support for OGC WFS. The release also includes various smaller improvements that enhance the developer experience.

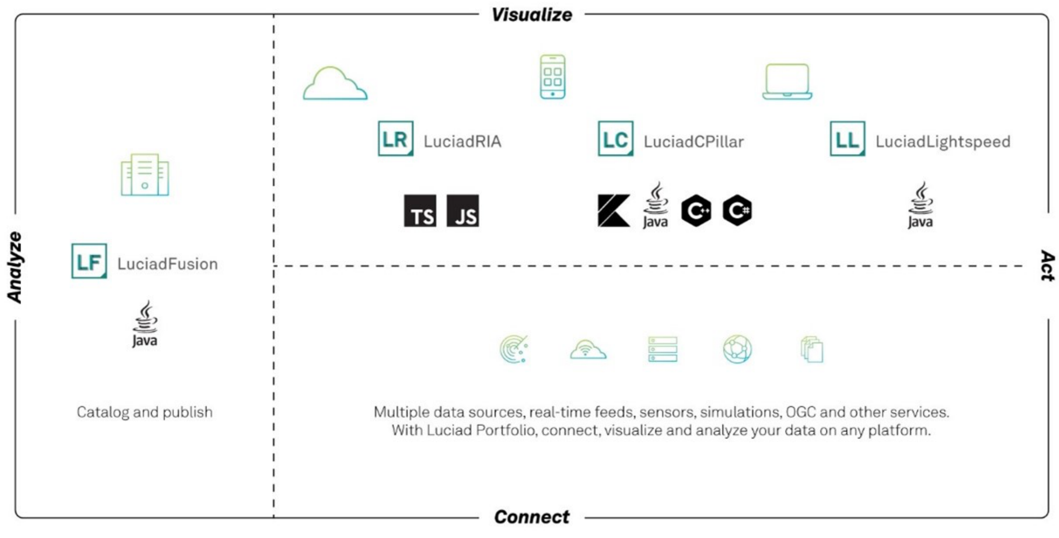
Visualize
LuciadRIA
LuciadCPillar
LuciadLightspeed
TS
JS
Java
Java
Analyze
LuciadFusion
Java
Catalog and publish
Act
Multiple data sources, real-time feeds, sensors, simulations, OGC and other services.
With Luciad Portfolio, connect, visualize and analyze your data on any platform.
Connect

Figure 1: The Luciad portfolio

Benefits of the new features

WebGPU replaces OpenGL

LuciadCPillar 2026.0 introduces support for WebGPU as the underlying graphics API for hardware-accelerated rendering, effectively replacing OpenGL for desktop and OpenGL ES on Android devices. This transition allows LuciadCPillar to leverage the latest advancements in web graphics technology, paving the way for enhanced performance and new features in future releases.

WebGPU benefits

OpenGL is based on a state machine model, which is less efficient for today's GPUs. WebGPU, on the other hand, uses a command buffer and pipeline-based approach, making it better-suited for modern hardware. Moreover, WebGPU abstracts similar modern APIs under one interface, ensuring consistent behavior across platforms. Specifically, WebGPU abstracts Vulkan on Android, Metal on iOS, and Direct3D 12 on Windows. This means that, for developers integrating LuciadCPillar via C#, there is easier integration with .NET. What's more, OpenGL ES is being phased out in favor of Vulkan on Android. Regardless, developers using LuciadCPillar on Android can rest assured that they can keep using LuciadCPillar for many years to come.

Minimal transition impact

WebGPU was introduced under the hood. This means that, except for a few well-documented upgrade considerations, this change has no impact on your code. You can find the upgrade considerations in the release notes.
The introduction of WebGPU means that the previous 2025.0 release of LuciadCPillar marks the end of the hardware-accelerated rendering based on OpenGL and OpengL ES. The 2025.0 version of LuciadCPillar will be actively maintained until the release of version 2028. Contact the Luciad Product Management team at product.management.luciad.gsp@hexagon.com if you plan to extend your maintenance on LuciadCPillar 2025 beyond 2028 so we are aware of your project.

Parameterized line and area styling and filtering

This release adds parameterized line and area styling and filtering to LuciadCPillar, completing the existing capability to style and filter points or plots. Parameterized styling and filtering is a unique concept throughout the products of the Luciad portfolio. It involves the use of style expressions to specify how to display features. LuciadCPillar evaluates these expressions on the GPU, on each frame, for an entire group of features. The main advantage of this approach over non-expression-based feature styling is that values in the style expressions can adapt continuously with almost no overhead. This means that you can efficiently perform style updates for many features.
The completed capability, as released with LuciadCPillar 2026.0 allows you to build style expressions using the parameters like color, size of icons, thickness of lines, and visibility to filter out features.

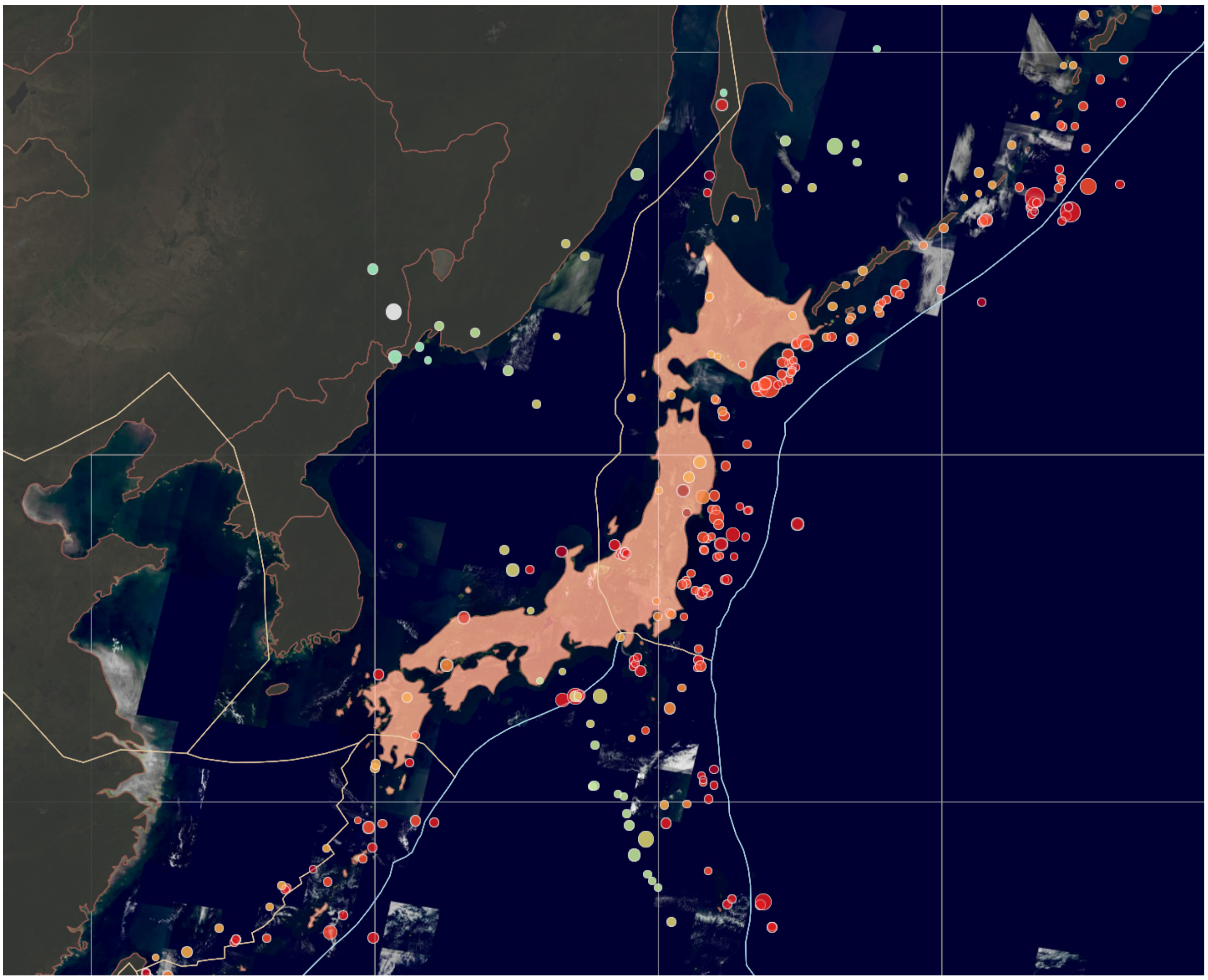

Figure 2: Areas and points with varying colors and scale depending on feature properties

To help you build those expressions, the StyleExpressionFactory offers building blocks as well as various arithmetic and control flow operations to combine the building blocks in more complex expressions.

Sample code/documentation to get you started

The article Parameterized feature styling offers a comprehensive overview of how to use parameterized styling and filtering. You can find an illustration of this feature in a dedicated sample titled "Parameterized Styling".

Extended data format support

This release adds support for three additional data formats: ECW, JPEG2000, and GML.

Decode and visualize ECW data

Hexagon's Enhanced Enhanced Compression Wavelet (ECW) technology is an image compression format designed specifically for handling large geospatial raster datasets, such as aerial imagery, satellite images, and orthophotos, in an efficient and scalable way.
ECW uses wavelet-based compression (similar to JPEG 2000) to achieve very high compression ratios while maintaining visually acceptable image quality. This makes it ideal for storing and streaming massive raster files

without overwhelming storage or network resources. The format is optimized for fast decompression and real-time display, which is crucial for GIS applications where users need to pan and zoom large images quickly.

LuciadCPillar now allows you to decode ECW files and visualize their content as raster data.

Decode and visualize JPEG2000 data

JPEG2000 is an advanced image compression format developed as a successor to the original JPEG standard. It uses a wavelet-based compression method, which allows for more efficient encoding and better image quality at higher compression ratios. Additionally, it supports both lossy and lossless compression, giving users flexibility to either preserve all image data or reduce file size while maintaining acceptable quality. This combination makes JPEG 2000 ideal for applications requiring high fidelity and scalability, such as medical imaging, geospatial data, and digital archiving.
LuciadCPillar now allows you to decode JPEG2000 files and visualize their content as raster data.

Decode and visualize GML data

The OGC GML format stands for Geography Markup Language, which is an XML-based encoding standard developed by the Open Geospatial Consortium (OGC). It is primarily used for representing geographical features and their properties in a structured, interoperable way.
LuciadCPillar now allows you to decode GML files and visualize their content as feature data.

Note that a Geography Markup Language (GML) file may contain multiple feature types. LuciadCPillar currently supports the simple features profile for GML files, which allows a single feature type in a model. It is possible to extract all feature types from a GML file by decoding them in separate models. The current support is documented with GmlModelDecoder.

Sample code/documentation to get you started

The following articles provide guidance on the newly added formats:

- Decode and visualize ECW data
- Decode and visualize JPEG2000 data
- Decode and visualize GML data

You can find an illustration of the LuciadCPillar data format support in the sample titled "Data Formats".

Extended OGC services support: WFS

The OGC WFS specification defines a standard interface for querying and manipulating geographic data. Queries can be formulated based on filters. The default exchange format for geographic features is GML, a vector data format. This means that the information can be retrieved without any prior interpretation or rendering, and with full georeferencing. This contrasts with, for instance, the OGC WMS interface, which renders maps on the server side and returns simple images to the client.
LuciadCPillar now allows you to discover and query geographic features on an OGC server. It offers a WFS client API, a simple framework for connecting to a WFS and retrieving feature data from it.

The versions 1.1.0 and 2.0 of the WFS specification are supported.

Sample code/documentation to get you started

There is a series of articles providing guidance on retrieving information from an OGC WFS service. You can start with Decode and visualize WFS data.
The sample titled "Data Formats" illustrates how to connect to OGC WFS services.

Other improvements

New WMS and WMTS media type

The WmsModelDecoder and WmtsModelDecoder now also support the image/jpgpng media type, which can be served by LuciadFusion. It offers a dynamic behavior allowing users to optimize both transparency and performance without manually choosing a format.

New WinUI sample for C#

We have added a new sample for C#, illustrating the use of the LuciadCPillar map in WinUI.

Prerequisites are upgraded

Please take note of the following updates:LuciadCPillar now targets the C++ 20 standard.The minimum gcc compiler version on Linux is now version 13.The minimum version of Visual Studio is now Visual Studio 2022 (v17.10).

Third-party upgrades

Several third-party libraries were updated for this release, because of known security vulnerabilities in older versions. You can find all details in the release notes.