

ORACLE FUSION MIDDLEWARE MAPVIEWER 11G

MAPVIEWER KEY FEATURES

- Component of Fusion Middleware
- Integration with Oracle Spatial, Oracle Locator
- Support for two-dimensional vector geometries stored in Oracle Spatial, as well as GeoRaster data and data in the Oracle Spatial topology and network data models
- Map rendering engine
- Map Builder tool
- Oracle Maps map cache and feature of interest servers
- Interactive querying and rendering of geospatial data
- Customizable styles, themes, and base maps
- “Slippy” map pan, zoom and slide
- XML, Java and JavaScript APIs
- Internet and Java standards-based deployment
- Certified with Oracle Application Server and Oracle WebLogic Server version 9 and higher
- Open Geospatial Consortium (OGC)-compliant Web Map Service (WMS) server

Oracle Fusion Middleware MapViewer provides powerful geospatial data visualization and reporting on Oracle Database. MapViewer provides web application developers with XML, Java and JavaScript APIs to integrate and visualize business data with maps. It deploys on an Internet and Java standards-based platform and integrates maps and spatial analysis into web and standalone applications. It provides: transparent and interactive querying and rendering of geospatial data; highly customizable styles, themes, and base maps; magnification-sensitive rendering of features; and “slippy” maps that pan, zoom and slide in any direction.

MapViewer Overview

MapViewer is a component of Oracle Fusion Middleware. It allows developers to incorporate maps and spatial analysis into business applications. MapViewer is a mid-tier toolkit that runs as a service inside an instance of Oracle Container for J2EE (OC4J) or other J2EE container. MapViewer version 11g is certified on Oracle WebLogic Server version 10 and higher.

MapViewer includes the following main components:

- A core rendering engine (Java library) named SDOVIS that performs cartographic rendering. A servlet is provided to expose the rendering functions to Web applications.
- A suite of application programming interfaces (APIs) that allow programmable access to MapViewer features. These APIs include XML, Java, PL/SQL, and an AJAX-based JavaScript API.
- A graphical Map Builder tool that enables you to create map symbols, define spatial data rendering rules, and create and edit MapViewer objects.
- Oracle Maps, which includes map cache and FOI (feature of interest) servers that facilitate the development of interactive geospatial Web applications.

The core rendering engine connects to the Oracle database through Java Database Connectivity (JDBC). It also reads the map metadata (such as map definitions, styling rules, and symbologies created through the Map Builder tool) from the database, and applies the metadata to the retrieved spatial data during rendering operations.

The XML API provides application developers with a versatile interface for submitting a map request to MapViewer and retrieving the map response. The

JavaBean-based API and the PL/SQL API provide access to MapViewer's rendering capabilities. The JavaScript API enables you to create highly interactive web applications that use the Oracle Maps feature of MapViewer.

The Map Builder tool simplifies the process of creating and managing map, theme, and symbology metadata in a spatial database.

Oracle Maps, built on core MapViewer features, uses a map tile server that caches map image tiles, and a feature of interest (FOI) server that streams live data out of a database to be displayed as interactive features on a map. You can use the AJAX-based JavaScript API with Oracle Maps to provide sophisticated mapping solutions. Oracle Maps also allows for advanced customization and querying capabilities.

The primary benefit of MapViewer is its integration with Oracle Spatial, Oracle Locator, and other Oracle Fusion Middleware components. MapViewer supports two-dimensional vector geometries stored in Oracle Spatial, as well as GeoRaster data and data in the Oracle Spatial topology and network data models. Oracle MapViewer is also an Open Geospatial Consortium (OGC)-compliant Web Map Service (WMS) server.

Prerequisite Software for MapViewer

To use MapViewer, you must have the following Java packages and Oracle products, with the release number listed or a later release:

- Oracle Application Server 11g or later, or a standalone version of Oracle Application Server Containers for J2EE (OC4J) Release 11g or later, or WebLogic Server 10.0 or later. These are available from the Oracle Technology Network at <http://www.oracle.com/technology/>
- Oracle Database with Spatial or Locator (Release 9i or later)
- Oracle Client (Release 9i or later), if you need to use JDBC Oracle Call Interface (OCI) features. Note that in general, the JDBC thin driver is recommended for use with MapViewer, in which case Oracle Client is not required.
- Java SDK 1.5 or later

MapViewer also supports the headless AWT mechanism in J2SE SDK, which enables MapViewer to run on Linux or UNIX systems without setting any X11 DISPLAY variable.

New MapViewer Release 11g Features

Secure Map Rendering based on a Web user's identity. Users with different roles or permissions will see different feature sets when viewing the same theme.

Web Feature Service (WFS) features can now be viewed through WFS themes. These themes support the parsing and caching of WFS capabilities, and the use of feature conditions and queries. WFS theme support also works with Oracle Maps; for example, you can display a WFS theme as an interactive feature of interest (FOI) layer.

Map Builder tool enhancements. The Map Builder tool now supports the creation of WFS themes and Annotation Text-based themes. The base map panel also supports identifying features (and a list of rendered themes) on mouse clicks.

Improved Nonspatial Data Provider Support. MapViewer now supports a default provider and format for applications to supply XML-based nonspatial data for thematic mapping. You can also specify columns from the nonspatial data set to be used in conjunction with an advanced style.

Multiple Rendering Styles for a Single Feature. You can now use multiple rendering styles when rendering a theme's features. For example, you can shade a polygon with a color style while also plotting a pie chart on top of it, without defining two themes. This is done using stacked styles in a theme's definition

Automatic Reduction of Repetitive Labels. You can now use the Map Builder tool to specify the No Repetitive Labels option in the base map properties, to cause features (such as road segments) with same name to be labeled only once

MapViewer Java Portlet Specification (JPS) Portlet. MapViewer now comes with a simple JSR 168-compliant Portlet that enables easy inclusion within an enterprise portal based on the Oracle Web Center technology.

Scale Ranges for Theme Labeling: In the context of a base map, you can now assign scale limits to its themes' labels. These scale limits control when a theme's features will display their label texts.

PDF Output. Full PDF map output support is provided. If you use PDF_URL or PDF_STREAM as the map format in your XML map request, MapViewer will generate vector PDF maps.

Text Style Enhancements. The TEXT style has been improved to support customizable spacing between letters. It also supports additional (vertical) alignment options when labeling linear features.

Custom Tags for Theme and Base Map Definitions. The XML definition of a theme or base map now supports application-specific attribute tags. You can use the Custom Tags option in the theme definition in Map Builder to specify tags and their values, which can be interpreted by your application but are ignored by MapViewer itself.

Getting Style Names Referred to in a Predefined Theme. The new <list_theme_styles> element enables you to get the names of styles referred to in a predefined theme.

Annotation Text. Support is provided for OpenGIS Consortium standard annotation text. Oracle Spatial in Oracle Database Release 11g supports storage of annotation text objects in the database, and MapViewer now supports displaying such annotation texts on a map.

Logging Mechanism Changes. A new logging mechanism based on Java logging is provided. You can also use the Oracle Application Server management console to customize how MapViewer logs things at run time.

Custom (External) Spatial Data Providers. MapViewer now supports rendering of geospatial data stored in non-Oracle Spatial repositories. This is achieved through a Custom Spatial Data Provider API, where you can implement an Interface that feeds your own (proprietary) spatial data to MapViewer for rendering. Note that you will still need an Oracle Database to manage the mapping metadata, such as styles and themes definitions.

User-Specified JDBC Fetch Size for Predefined Themes. You can now specify a nondefault row fetch size on a theme, by setting the Fetch Size base map property with the Map Builder tool. MapViewer can use this value when fetching theme features from the database. Specifying an appropriate value can make performance tuning easier in certain situations.

Heat maps. MapViewer now supports heat maps, which are two-dimensional color maps of point data sets.

Scalable styles. MapViewer now supports scalable styles. A scalable style (such as a MARKER or LINE style) uses real-world units such as meter or mile to specify its size and other dimensional attributes; however, at run time MapViewer automatically scales the style so that the features rendered by the style always show the correct size, regardless of the current map display scale.

URL request to get sample image for a style. You can now issue a simple URL request to the MapViewer server and get back a sample image of any style that you specified in the URL. This is useful if you want to build a custom map legend.

New array types (MV_XXXXLIST). MapViewer uses the SQL array types MV_STRINGLIST, MV_NUMBERLIST, and MV_DATELIST, which support array-type binding variables that might exist in some predefined themes. In some situations, you will need to create these types.

New Oracle Maps Features

External tile layers. The Oracle Maps JavaScript client can now display map tiles rendered directly by an external map tile server without caching the tiles with the MapViewer map tile server.

Informational tips for navigation panel. Applications can now define mouseover informational tips or labels for map zoom levels. The informational tips are displayed when the user moves the mouse over the navigation panel. The user can then zoom to a selected zoom level by clicking on the corresponding info tip.

Built-in toolbar and distance measurement. Applications can now use a built-in distance measurement tool to measure distance on the map. The built-in toolbar provides an easy graphic user interface for accessing utilities such as the redline tool, rectangle tool, circle tool, distance measurement tool, and any user-defined capabilities.

Automatic Determination for Whole Image Theme Display. Displaying a theme-based FOI layer as a whole image may greatly improve the application performance, but it may be difficult for application developers to determine when to display a theme as a whole image theme. However, you can now choose to let MapViewer

make the determination automatically.

Automatic long tile administrative request recovery. Long running tile admin requests that are interrupted due to Fusion Middleware or MapViewer shutdown will be able to resume automatically after MapViewer is restarted. (You do not need to do anything to enable this feature, other than creating the new database view USER_SDO_TILE_ADMIN_TASKS if it does not already exist.

Wraparound map display. Applications can now display a map in the wrap-around manner. When the map is displayed in this manner, the map wraps around at the map coordinate system boundary horizontally and therefore can be scrolled endlessly.

Enhanced redline tool. The redline line tool can now be used to create polyline, polygon, and point geometries. The redline line tool also supports an editing mode, in which you can move an existing redline point or line segment, remove a redline point or line segment, or add a redline point or line segment programmatically.

Individual theme feature highlighting. Applications can enable the user to select and highlight individual theme features (FOIs) by clicking the mouse on the features.

JDBC Theme-Based FOI. Oracle Maps now supports client side dynamically constructed JDBC theme-based FOI layers

Simplified Dynamic BI Data Injection and Visualization. Business Intelligence applications can now visualize application generated nonspatial attribute data on the map through the combined use of a nonspatial data provider and theme-based FOIs.

Improved Information Window. The positioning, styling, and sizing of the information window have been improved. The Oracle Maps client now can place the information window at the optimal variable position relative to the specified map location. As the result, the map does not to be panned in order to make the information window visible inside the map. In addition, you can specify tabs for the information window.

Enhanced Map Decoration. The client now supports multiple collapsible map decoration components that can be positioned at arbitrary positions inside the map container. Map decoration can now be dragged inside the map container.

Flexible Placement and Visibility for Navigation Panel and Scale Bar. The navigation panel and the scale bar can now be placed inside a map decoration component, which can be displayed or hidden and can be placed at a position of your choice inside the map container.

Polygon Theme-Based FOI Layer Labeling. Applications can now choose whether to label the polygon features of a polygon theme-based FOI layer.

Image-Less Polygon Themes with FOI Layers. You can now base a feature of interest (FOI) layer on an image-less polygon theme, causing FOI images not to be rendered if the theme is already rendered as part of the base map. This feature can result in much faster performance with polygon layers.

FOI Layer Automatic Selection and Highlighting. You can now associate a filtering geometry with any predefined theme-based FOI layer so that only the features that fall inside the filtering geometry are rendered on the map. This feature is part of the new support for proximity and within-distance mapping. It can be used with the selection tools (circle, rectangle, or polygon) to implement theme feature highlighting.

Client-Side Construction of Geodetic Geometries Based on Earth Distance Parameters. You can now construct the following type of geometries in geodetic coordinate systems with parameters based on Earth distance: a circle polygon geometry specified by its center and radius, a rectangle with its height and width specified, and a point geometry at the specified distance and bearing from the start point. These geometries, especially the first two, can be used for implementing proximity and within-distance type mapping.

Animated Loading Icon for Maps and Themes. The Oracle Maps client now displays an animated icon during the loading of a base map or a theme. This is especially useful for providing visual reassurance to users with maps and themes that take a long time to load.

User-Defined FOI Customizations. The JavaScript API now provides methods for applications to modify the geometry representation and rendering style of an already rendered user-defined FOI, as well as the custom marker image for a user-defined point FOI.

Prompt Mode for Marquee Zoom Tool. The new prompt mode provides enhanced user control over marquee zoom operations. Prompt mode zooms the map when the user clicks on the marquee zoom rectangle, which eliminates the possible problem of accidental zooming associated with "continuous" mode.

Mouse Cursor Customization. Applications can now customize the appearance of the mouse cursor when the cursor is over different map components, such as map tiles, FOIs, and map decorations.

Error Reporting. Previously, all error messages thrown by the Oracle Maps client were displayed as browser alerts. Now applications can customize how the error messages are handled by using a custom error handler.

New Tutorials. Many new Oracle Maps tutorials illustrate the new features. Find them at: <http://host:port/mapviewer/fsmc/tutorial/demos.html>

MapViewer saves money by reducing development time through tight integration with the ADF environment. MapViewer increases the value of location data in Oracle Database by providing a tightly coupled tool to visualize these data. It can also be used as a powerful "mash-up" component to integrate standards-based and commodity mapping services (e.g.: OGC Web Services, GoogleMaps, Virtual Earth, Map24, Where2GetIt, etc.) with application data stored in Oracle Database and Oracle BI EE sources. MapViewer improves applications summarizing complex business and geographic data and relationships in an easily understood, universally recognized format - a map.

KEY BENEFITS

- Summarize complex business and geographic data and relationships
- “Mash-up” standards-based and commodity mapping services with application data
- Reduce development time

RELATED PRODUCTS AND SERVICES:

MapViewer is integrated with a variety of Oracle tools and third party applications:

- Oracle Business Intelligence Enterprise Edition
- Oracle Application Express
- Oracle JDeveloper ADF Data Visualization
- Fusion Middleware products: Oracle WebCenter, BAM, SOA Suite
- Oracle applications: Utilities, Communications, Retail, HR, Service
- Many ISV tools and applications

For more information go to <http://otn.oracle.com> and search on “MapViewer”



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. 0109