

# Oracle AutoVue 20.1

---

## AutoVue Hotspots API Guide


**Oracle Corporation Headquarters**

500 Oracle Parkway  
Redwood Shores, CA  
94065  
U.S.A

**North America Inquiries**

+1.514.905.8400  
Toll Free: 1.800.363.5805

**EMEA Inquiries**

Germany / Eastern Europe:  
+1.514.7353219  
UK/Nordic:  
+44.870.8768761  
France / Southern Europe / MEA:  
+35318031307

**APAC Inquiries**

China / Taiwan / Hong Kong / Korea /  
India:  
+1.514.594.3366  
+86 10 6535 6228  
+86 135 0129 0319

**South Asia / Australia Inquiries**

+61294911304

autovuesales\_ww@oracle.com

<http://www.oracle.com/us/products/applications/autovue/index.html>

**Copyright © 1989, 2011, Oracle and/or its affiliates. All rights reserved.**

Portions of this software Copyright 1996-2007 Glyph & Cog, LLC.

Portions of this software Copyright Unisearch Ltd, Australia.

Portions of this software are owned by Siemens PLM © 1986-2008. All rights reserved.

This software uses ACIS® software by Spatial Technology Inc. ACIS® Copyright © 1994-1999 Spatial Technology Inc. All rights reserved.

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

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

**U.S. GOVERNMENT RIGHTS**

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

# Table of Contents

---

1	Preface.....	4
	Audience.....	4
	Documentation Accessibility.....	4
	Accessibility of Code Examples in Documentation .....	4
	Accessibility of Links to Extern Web Sites in Documentation .....	4
	Deaf/Hard of Hearing Access to Oracle Support Services.....	4
	Related Documents .....	4
2	Introduction .....	5
	Text Search Hotspots .....	5
	Web CGM Hotspots.....	5
3	AutoVue Behaviour on Hotspots.....	6
4	AutoVue API for External System Interaction .....	6
5	AutoVue Hotspots API .....	7
	Define Hotspots .....	7
	Hotspots Handler Types .....	7
	Hotspots Definition.....	7
	Perform an Action on a Hotspot .....	8
	Hotspots Actions .....	8
6	Interactions with Hotspots from JavaScript .....	8
7	Hotspots Sample.....	9
8	Feedback .....	11

# 1 Preface

The *AutoVue Hotspots API* Guide provides information on how to build on the hotspots capability of AutoVue.

For the most up-to-date version of this document, go to the AutoVue Documentation Web site on the Oracle Technology Network at <http://www.oracle.com/technetwork/documentation/autovue-091442.html>.

## ***Audience***

This document is intended for Oracle partners and third-party developers (such as integrators) who want to implement their own integration with AutoVue. This guide serves as a good starting point for developers and professional services to become more familiar with the hotspots.

## ***Documentation Accessibility***

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

## ***Accessibility of Code Examples in Documentation***

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

## ***Accessibility of Links to Extern Web Sites in Documentation***

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

## ***Deaf/Hard of Hearing Access to Oracle Support Services***

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

## ***Related Documents***

For more information, see the following documents:

- *AutoVue API Programmer's Guide*
- *VueBean Javadocs*

## 2 Introduction

AutoVue 20.1 includes a hotspot capability that allows system integrators to create links between objects in AutoVue's data model and objects in an external system. With this feature, a system can be built that integrates AutoVue tightly into other applications. By clicking on sections of text in a document in AutoVue, an action is triggered and/or information displays in other applications.

AutoVue provides the following capabilities around hotspots:

- Hotspot definitions
- Data connection information defining AutoVue hotspots linked to external objects
- Tooltip to display on the hotspots defined in AutoVue
- Customization for hotspot selection notification
- Customization for available actions for selected hotspot

The following types of documents can represent a hotspot in AutoVue:

- Text in 2D and EDA documents (based on AutoVue's text search capability)
- Web CGM format

**Note:** It is possible to extend the AutoVue applet using the `VueAction()` method to implement a hotspot action. Refer to the *AutoVue API Programmer's Guide* for a `VueAction()` hotspot example.

### ***Text Search Hotspots***

Hotspot support in 2D and EDA documents is based on regular expressions filtering graphical text strings based on AutoVue's text search:

- You can use regular expressions in the hotspot definition. Since AutoVue uses the Java library, it relies on Java's regular expression guidelines. For more information, refer to the Java regular expression guidelines at: <http://java.sun.com/developer/technicalArticles/releases/1.4regex>. Hotspots rely on AutoVue's text search capability. AutoVue's text search behaviour varies depending on the file format. For example, for some file formats a string could be returned as a sequence of characters, while for others, a string could be returned as a string. Ensure that you construct the regular expression in the hotspot definition based on AutoVue's text search ability for the format.
- Hotspots are supported on the base file only. They are not supported on overlaid files.
- In Compare and Cross-Probe modes, hotspots only function on the base file window.

### ***Web CGM Hotspots***

In Web CGM files, hotspots are defined in the native file. The hotspot information contains three attributes:

- Name
- ID
- URI

External systems can interact with these hotspots using the `VueBean` API using a given name. AutoVue matches the name to the ID property of the hotspot. If this fails, AutoVue matches the name to the Name property in order to highlight a specific hotspot.

### 3 AutoVue Behaviour on Hotspots

AutoVue handles the following user interactions around hotspots:

- When the mouse cursor is on top of a hotspot, a visual hint is displayed to user to indicate that the hotspot can be interacted (i.e. can be clicked). A tooltip is displayed to show its description.
  - The hotspot tooltips have the following priority ranking in the stack of tooltips precedence:
    - Markup tooltip
    - Measurement tooltip
    - Hotspot tooltip
    - EDA Entity Information tooltip
    - Hyperlink tooltip
- When a user clicks on a hotspot on the display, a notification is fired to the external system with the information identifying the clicked hotspot and the mouse action (Click vs. Double-Click) and keyboard modifiers (Ctrl, Shift, Alt)
  - The mouse click and double-click on a hotspot fires the notification to an external application following these precedence rules:
    - Markup: Consumes the click / double-click.
    - Measurement: Consumes the click / double-click.
    - Hotspot: Notifies the external application but does not consume the click / double-click and allows the subsequent layers to handle the click / double-click as well.
    - Hyperlink: Does not consume the click / double-click.
    - EDA Entity selection, 3D Entity selection, Entity properties on double-click, and so on.
- When a user right-clicks on the hotspot, a menu displays with the available actions on this hotspot pre-defined by the integrator. When the user clicks on one of the menu items, a notification is fired to the external system with the information identifying the clicked hotspot and the action selected by the user.

### 4 AutoVue API for External System Interaction

An external system can call the AutoVue API for manipulating hotspots from the following user actions:

- Highlight (Multiple Selection, Add/Remove)
  - Text Highlight as used in text search.
  - 2D Entity Highlight for Web CGM format.
- Zoom to a hotspot, or the hotspots associated with a specific external object.
- Browse the hotspots associated with a specific external object using **Zoom Previous/Zoom Next**.

**Note:** When a user selects a hotspot, all hotspots associated with the same external object may be selected by using the highlight mechanism provided above.

## 5 AutoVue Hotspots API

There are two methods in the `jVue` class that handle the hotspots:

- `setHotSpotHandler()` to define hotspots
- `performHotSpot()` to perform an action on a hotspot

### Define Hotspots

`setHotSpotHandler (final String definitionType, final String definitionKey, final String Definition)`

This method sets the hotspot handler for a given hotspot definition. This should typically be called before opening the file. It initializes hotspots in the files opened in AutoVue based on external application data.

Parameter	Description
<i>definitionType</i>	The hotspot definition type. Specify if hotspot is WebCGM hotspot or a text search hotspot. See <a href="#">Hotspots Handler Types</a>
<i>definitionKey</i>	The hotspot definition key. This is the identifier for the hotspot.
<i>definition</i>	A string separated by semicolons specifying hotspot definition parameters. For example: name1 = value1; name2 = value2.

### Hotspots Handler Types

The hotspot definition types supported in `setHotSpotHandler()` are:

DEFINITION_TYPE_NATIVE	Native Web CGM hotspots
DEFINITION_TYPE_TEXT	Text search hotspots

### Hotspots Definition

The hotspot definition parameters supported in the key-value string parameter (*definition*) of the method `setHotSpotHandler()` are:

DEFINITION_TOOLTIP	The tooltip displayed when user hovers mouse over a hotspot defined by the handler.
DEFINITION_ONINIT	The java script method to call when page is loaded and ready to interact.
DEFINITION_FUNCTION	The JavaScript function to call when user performs an action on the hotspot.
DEFINITION_ACTIONS	Popup actions to show when user right-clicks on a hotspot
DEFINITION_COLOR	The highlight color to use when user hovers the mouse cursor over a hotspot
DEFINITION_REGEX	Regular expression to use only in Text Search Hotspot handlers.
DEFINITION_MATCHCASE	Whether to handle case sensitivity in Text Search Hotspot handlers only.

## Perform an Action on a Hotspot

`performHotSpot` (final String *definitionKey*, final String *hotspotKey*, final String *action*, final String *params*)

Perform a hotspot action on the given hotspot. This method should be called during the file session when the hotspots have been already initialized (only after the external application notifies that hotspots have been initialized in the file).

Parameters	Description
<i>definitionKey</i>	The hotspot definition key (the hotspot identifier) provided at creation.
<i>hotspotKey</i>	The hotspot property key string found based on the definition key.
<i>action</i>	The action to perform on the hotspot. Refer to <a href="#">Hotspots Actions</a> .
<i>params</i>	A string separated by semicolons specifying hotspot action parameters. For example: name1 = value1; name2 = value2.

### Hotspots Actions

The hotspot actions supported in `performHotSpot()` and their arguments are:

Action Name	Description	Arguments
HIGHLIGHT	Perform a highlight	<b>HOTSPOT_COLOR:</b> The color for a highlight to add (RGBA Format). If this argument is not provided, the action is interpreted as a Highlight Removal.
ZOOMTO	Zoom to all hotspot instances	None
ZOOMNEXT	Zoom to the next hotspot instance	None
ZOOMPREV	Zoom to the previous hotspot instance	None

## 6 Interactions with Hotspots from JavaScript

The following is a code prototype for a custom JavaScript function call to initialize hotspots when the file/page loads:

```
initialization_script(String definitionKey)
```

The following is a code prototype for a custom JavaScript function call when a user interacts with hotspots:

```
notification_script(String definitionKey, String hotspotKey, String action, int keyModifiers, String properties)
```

`action` may be a custom action sent during the definition of the hotspot handler (RMB actions) or one these two predefined actions:



OnHotSpotClicked	To send when user clicks on the hotspot
OnHotSpotDoubleClicked	To send when user double-clicks on the hotspot

properties that could be sent to the external application notification script are:

PROPERTY_ID	ID of Native WebCGM Hotspots
PROPERTY_NAME	Name of Native WebCGM Hotspots
PROPERTY_URI	URI of Native WebCGM Hotspots

## 7 Hotspots Sample

The following hotspots example shows how the setHotSpotHandler() and performHotSpot() methods are implemented to add hotspotting capability to AutoVue. This example only adds one definition, but it is possible to add multiple definitions.

1. Initialize the hotspots with the ONINIT applet parameter. This parameter is used to call the onAppletInit() method after the AutoVue applet has initialized.

**Note:** If a newly added definition key already exists, then the existing definition is replace by the new one.

```
<PARAM NAME="ONINIT" VALUE="onAppletInit();">

function onAppletInit() {
    var handlerStr = "DEFINITION_REGEX=AutoVue; DEFINITION_TOOLTIP=AutoVue 2D
        Professional";
    // The following function is called once when AutoVue is ready to interact with a hotspot.
    handlerStr += ";DEFINITION_ONINIT=onHotSpotInit";
    // The following function is called each time a hotspot is fired.
    handlerStr += ";DEFINITION_FUNCTION=onHotSpot; DEFINITION_ACTIONS=Menu1,
        Menu2";
    color = ((128 & 0xFF) << 24) | ((0 & 0xFF) << 16) | ((0 & 0xFF) << 8) | ((255 & 0xFF) << 0);
    handlerStr += ";DEFINITION_COLOR=" + color;

    //The following call sets up the hotspot definition.
    window.document.applets["JVue"].setHotSpotHandler("DEFINITION_TYPE_TEXT", "AV2D",
        handlerStr);
}
```

2. Method onHotSpotInit() is called for each definition when the current page is loaded and ready for hotspot interactions. Note that the method name should be exactly the same as the one specified in the hotspot definition DEFINITION\_ONINIT in step 1.

```
function onHotSpotInit(hotspotDefinitionKey) {
    alert("HotSpot definition initialized: " + hotspotDefinitionKey);
}
```

3. The following onHotSpot() method is invoked when a hotspot is fired when the user either clicks on the hotspot or by selecting one of the Hotspot menu items defined in variable DEFINITION\_ACTION in step 1.

```

function onHotSpot(hotspotDefinitionKey, hotspotKey, action, modifiers, properties) {
  if (equalsIgnoreCase(action, "onHotSpotClicked")) {
    alert("User clicked on hotspot: " + hotspotKey);
  } else if (equalsIgnoreCase(action, "onHotSpotDoubleClicked")) {
    alert("User double clicked on hotspot: " + hotspotKey);
  } else if (equalsIgnoreCase(action, "Menu1")) {
    alert("User Peformed Menu1 action: " + hotspotKey);
  } else if (equalsIgnoreCase(action, "Menu2")) {
    alert("User Peformed Menu2 action: " + hotspotKey);
  }
}

```

Note that the method name should be exactly the same as the one specified in the hotspot definition DEFINITION\_FUNCTION in step 2. The onHotSpotClicked and onHotSpotDoubleClicked methods are predefined keys when the user clicks on the hotspot.

4. The following code performs specific actions on the clicked hotspot such as Highlight Zoom and so on.

```

// Highlight the "AutoVue" hotspot, "AV2D" is the definition key.
// Color : alpha | red | gree | blue
params = "HOTSPOT_COLOR=" + (((128 & 0xFF) << 24) | ((255 & 0xFF) << 16) | ((255 & 0xFF) << 8) | ((0 & 0xFF) << 0));
window.document.applets["JVue"].performHotSpot("Highlight", "AV2D", "AutoVue", params);

// To clear the hotspot highlight simply set the params (color) to null.
window.document.applets["JVue"].performHotSpot("Highlight", "AV2D", "AutoVue", null);

// To clear the definition highlights, set the hotspot key to null.
window.document.applets["JVue"].performHotSpot("Highlight", "AV2D", null, null);

// To clear all hotspot highlights, set the definition key to null.
window.document.applets["JVue"].performHotSpot("Highlight", null, null, null);

// Zoom to the next "AutoVue" hotspot.
window.document.applets["JVue"].performHotSpot("ZoomNext", "AV2D", "AutoVue", null);

// Zoom to the previous "AutoVue" hotspot.
window.document.applets["JVue"].performHotSpot("ZoomPrev", "AV2D", "AutoVue", null);

```

## 8 Feedback

We appreciate your feedback, comments or suggestions. Contact us by e-mail or telephone. Let us know what you think.

For any questions regarding a particular class or method, please contact Oracle Customer Support or post your question to the My Oracle Support AutoVue Community Web site. .

**General Inquiries:**

Telephone: +1.514.905.8400 or +1.800.363.5805

E-mail: [autovuesales\\_ww@oracle.com](mailto:autovuesales_ww@oracle.com)

Web Site: <http://www.oracle.com/us/products/applications/autoVue/index.html>

**Sales Inquiries:**

Telephone: +1.514.905. 8400 or +1.800.363.5805

E-mail: [autovuesales\\_ww@oracle.com](mailto:autovuesales_ww@oracle.com)

**Oracle Customer Support:**

Web Site: <http://www.oracle.com/support/index.html>

