

VueLink for SharePoint

***An Interface between
AutoVue Web Edition and
Microsoft SharePoint Server***

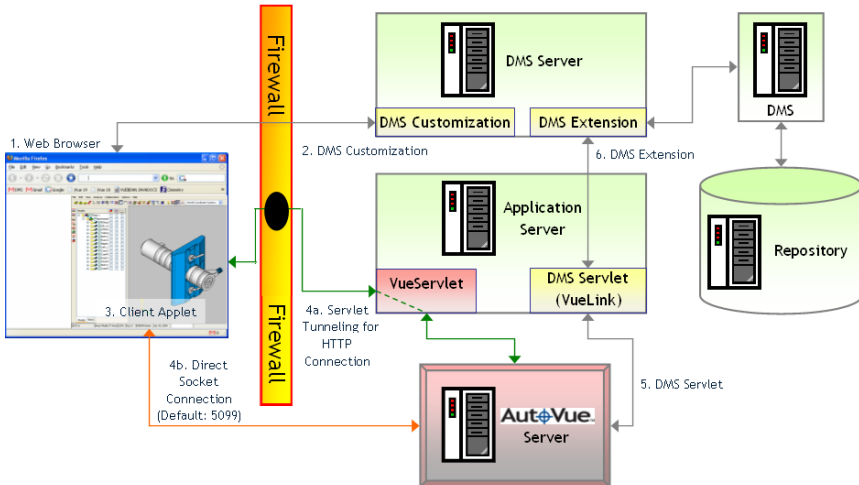
System Administrator Manual

Contents

INTRODUCTION	1
SYSTEM REQUIREMENTS	2
INSTALLATION.....	3
Installation Prerequisites.....	3
Automated Installation.....	3
Manual Installation.....	16
SharePoint Customization.....	16
SharePoint Extension	18
VueLink Servlet and Web Application	19
Configuration	19
Verification	19
CONFIGURATION.....	20
Setting DMS Launch Parameter in vue.aspx	20
Manual Steps for Customizing core.js	20
Setting AutoVue Applet Parameters in vuelink.properties	22
Enabling Markup and Metafile Functionality for New Sites.....	23
Enabling Viewing Files using Web Parts.....	24
Configuration and Support for XRef	26
Supported Mechanism.....	26
VERIFICATION	30
Verifying that VueLink is Running Properly.....	30
Running VueLink in Debug Mode	31
FEEDBACK	33
General Inquiries	33
Sales Inquiries	33
Customer Support.....	33

Introduction

The VueLink Servlet allows AutoVue Web Edition to communicate with Windows SharePoint Services/SharePoint Portal Server (SharePoint) using the standard HTTP protocol. The following figure and steps show a typical configuration for integration between AutoVue and SharePoint.



- 1 You must log on to the Microsoft SharePoint Server through a web browser such as Microsoft Internet Explorer.
- 2 With the Customization in place, clicking on a file opens a menu that lets you select **View in AutoVue**.
- 3 Clicking this option launches the AutoVue applet; you can view the file inside the web browser window.
- 4 Depending on the AutoVue configuration, the AutoVue applet communicates with the AutoVue Server either through servlet tunneling for HTTP connection or through direct socket connection.
- 5 AutoVue Server then communicates with the VueLink servlet using standard HTTP connection.
- 6 With DMS Extension installed on the server, the VueLink servlet can communicate with SharePoint to handle any request made by AutoVue Server such as file 'fetching'.

System Requirements

Note: For the most up-to-date list of system requirements, and for a list of known issues, see the release notes available in the **etc** subfolder of the VueLink distribution.

Server

- AutoVue Web Edition v19.2c1 (and any Service Packs for v19.2)

Note If you have installed AutoVue Web Edition v19.2, you must copy 'jvue.jar' from the AutoVue installation directory to the 'jVue' folder in your VueLink deployment on the application server.

- Microsoft Office SharePoint Server 2007 or Windows SharePoint Services 2007
- J2EE Web Application Server/Servlet Engine (e.g. Jakarta Tomcat 5.5.20)

Installation

Installation Prerequisites

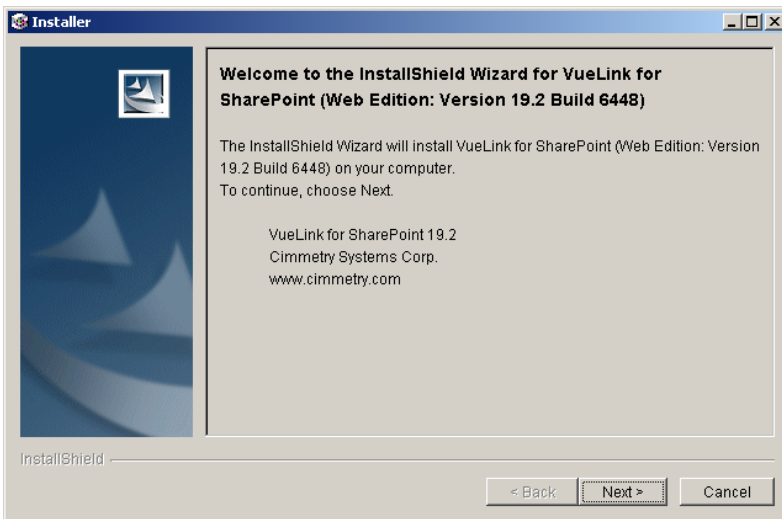
Before integrating AutoVue with SharePoint, ensure SharePoint, AutoVue Web Edition, and the application server (e.g. Tomcat) are properly installed and configured on your system according to the manufacturer's instructions. It is best to test either SharePoint or AutoVue independently to verify that the installation has been successful and that all functions are available and produce the expected results before integrating.

Automated Installation

The VueLink Installer for SharePoint installs the VueLink Servlet, and customizes and configures SharePoint and AutoVue Web Edition.

Note If you have a previous copy of VueLink for SharePoint installed, uninstall it before proceeding with the new installation.

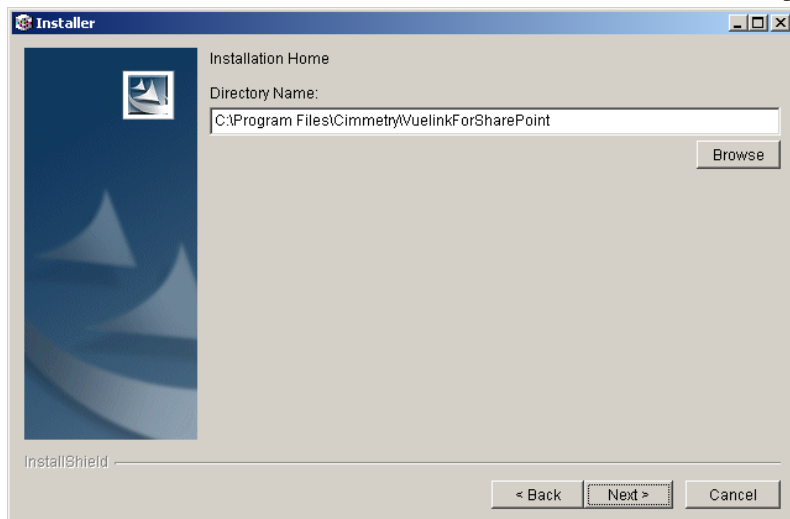
- 1 Run **setupwin32.exe** found on the Distribution CD in /AutoVue_Web_Edition/vuelink_setup/win32. The Installer starts installation of VueLink for SharePoint.



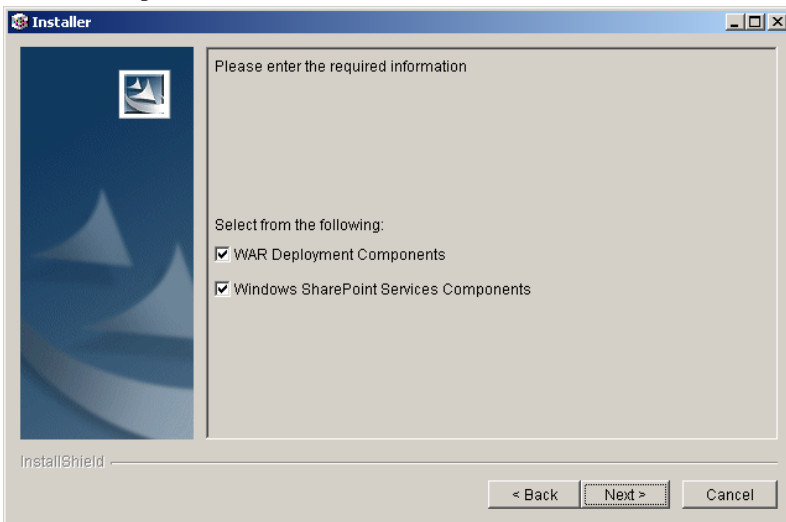
- 2 Read the VueLink License Agreement, select the option to agree to the terms, and click **Next**.



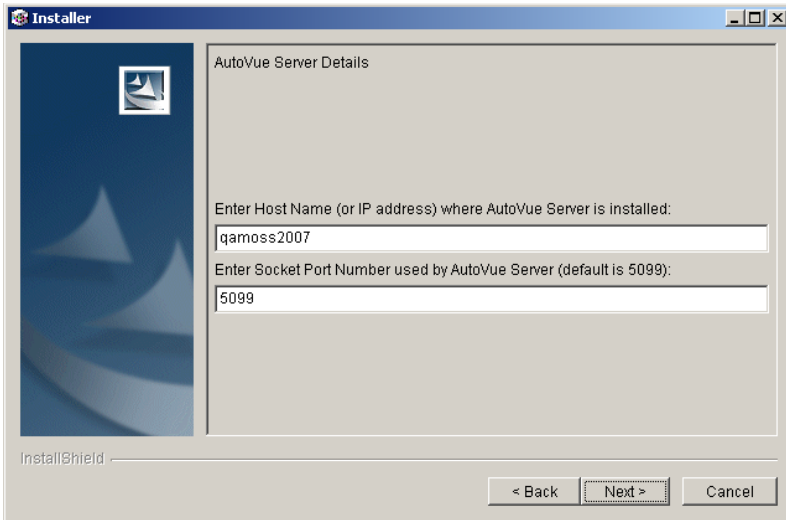
- 3 Enter the Installation home for VueLink for SharePoint. Setup creates a copy of all files installed or updated as part of the installation in this directory. These files can be referred to later in case the installation has to be updated.



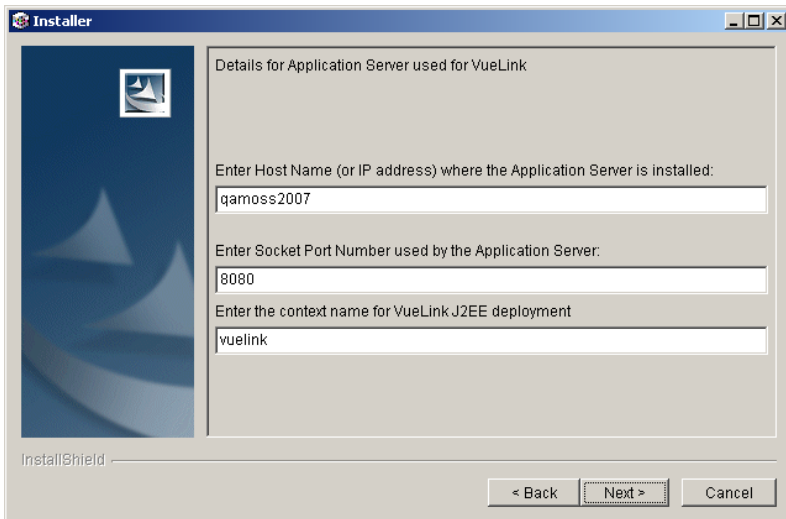
- 4 Choose the components that need to be installed:
- **WAR Deployment Components** installs VueLink components and deploys VueLink components in the application server. Note that this component needs a Servlet Engine or application server that is compatible with J2EE specification. The `vueLink.war` also exists in VueLink for SharePoint home directory in the `vueLink4j2ee` folder after installation. It can be deployed manually later on.
 - **Windows SharePoint Services Components** installs Cimmetry web services and add customization to components inside SharePoint.
- You may choose to install both components. The following steps deal with one component at a time.



- 5 Since the server part of AutoVue Web Edition is needed for every component, the Installer then prompts for the details of AutoVue Server. Enter the hostname of the machine running the AutoVue Server and the socket port of AutoVue Server.

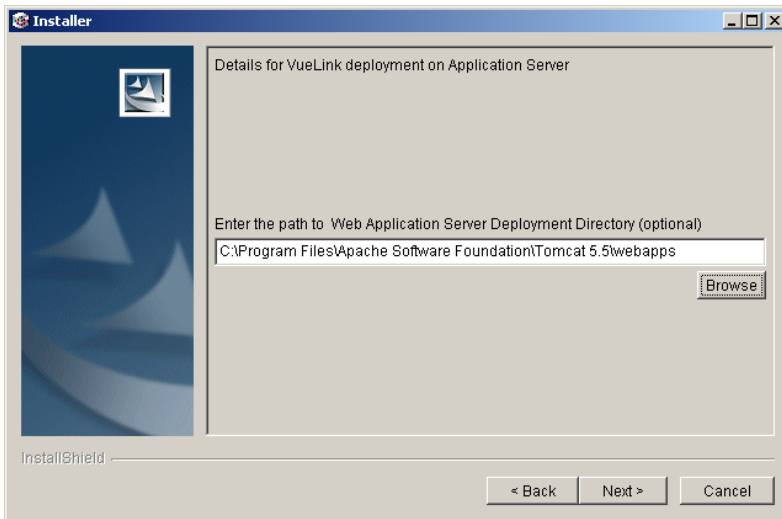


- 6 Enter the hostname, socket port, and the desired context name for VueLink for SharePoint.

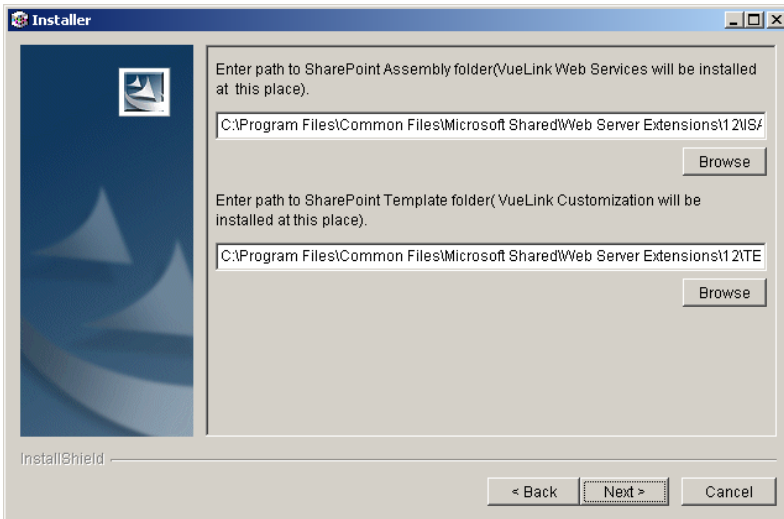


7 Enter the path to the deployment location of the application server.

Note The Installer installs application server components in the installation home directory and only deploys it under application server if the path to its deployment directory is given. If the application server is capable of hot deployment then proceed after entering requested information. Otherwise, make sure that the application server is not running. If it is, stop the application server before proceeding.

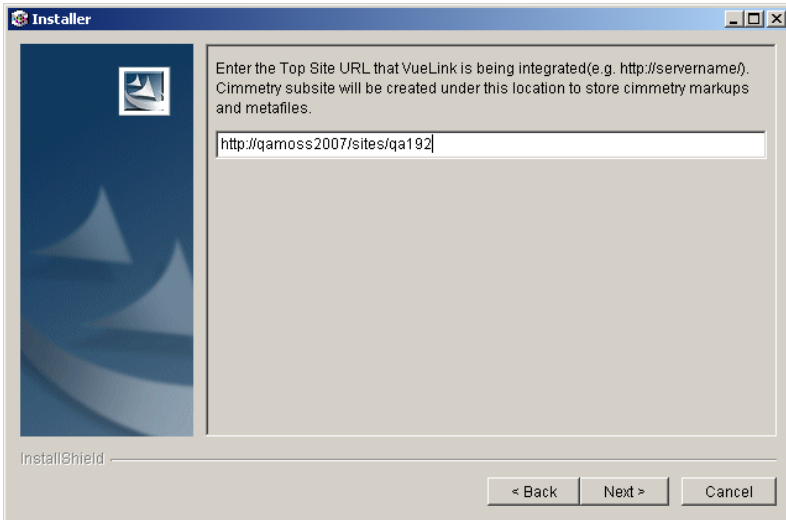


- 8 If you choose to install Windows SharePoint Services Components, the Installer prompts for SharePoint Assembly and Template folders.

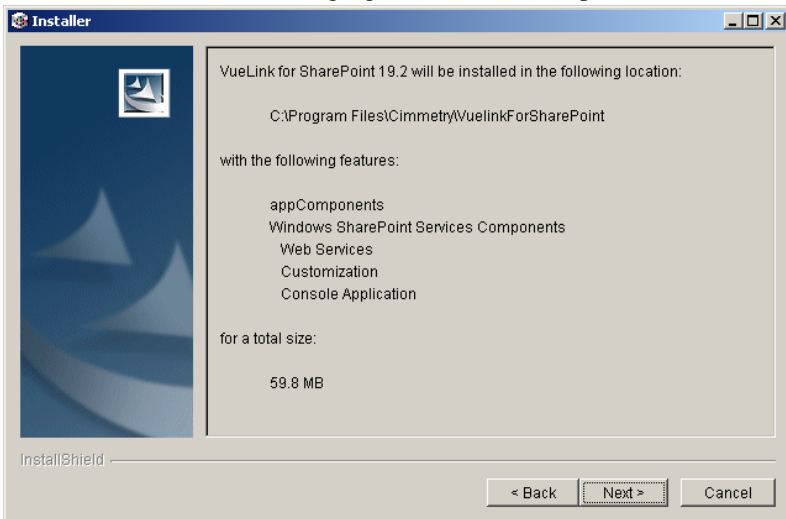


- 9 The Installer prompts for the URL of Top Site with which VueLink is being integrated. The Installer automatically creates a Cimmetry subsite for storing markups and metafiles at this location. You can also run this tool later on to create a Cimmetry subsite inside other top sites.

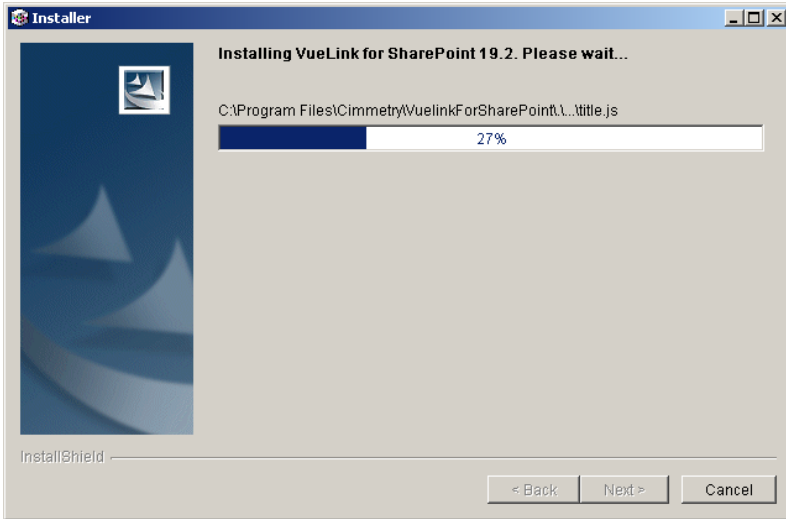
If you do not enter a URL, the Cimmetry tool is not invoked and a Cimmetry site is not created automatically.



10 The Installer summarizes the installation options, and a panel shows the information. The following figure shows an example:

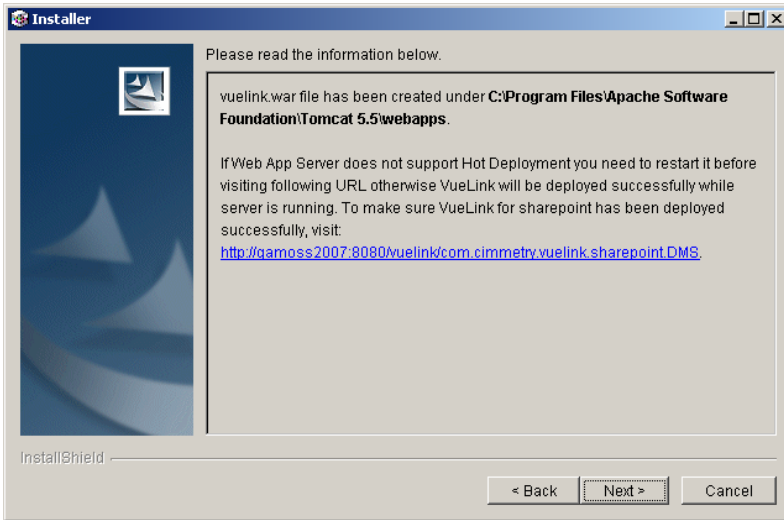


- 11 Click **Next** to continue installation. Wait until the Installer completes creating and updating files.

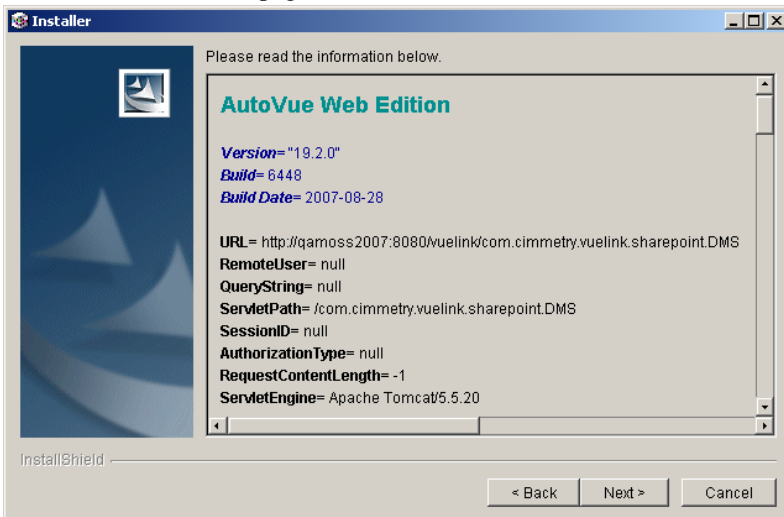


- 12 The Installer creates a log of the installation, which is stored in the installation folder.

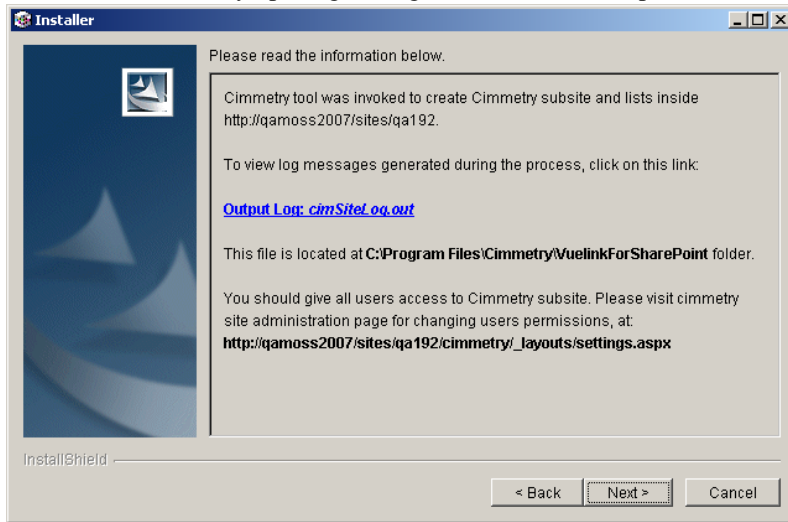
- 13 If you have chosen to install the WAR Deployment Components and you have provided the Installer with deployment directory, the Installer deploys `vueLink.war` and displays the link to the VueLink servlet test page.



If you click the link shown in the previous figure, the VueLink Test page opens inside the Installer window, if the application server is running. If it is not running, a message displays telling you that the Installer cannot connect to the VueLink test page.

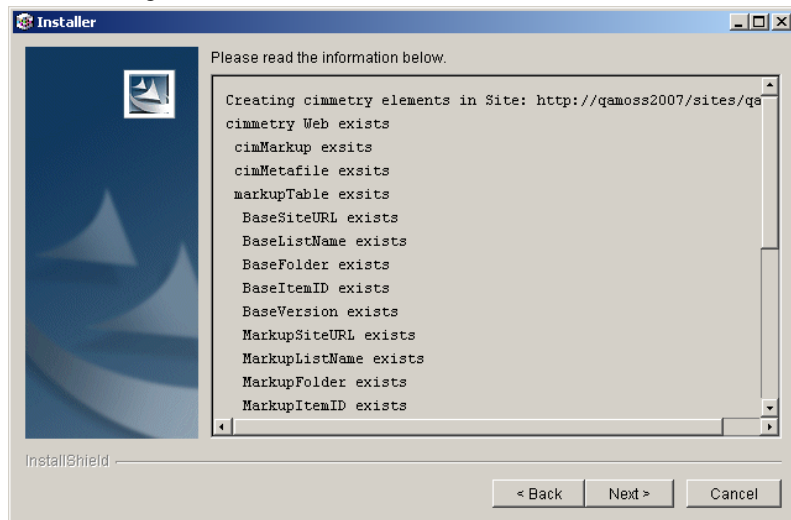


- 14 If you select Top Site URL for Cimmetry tool, the following page appears at the end of installation. You can check the content of its log file by clicking the blue link or by opening the log file at the location specified.

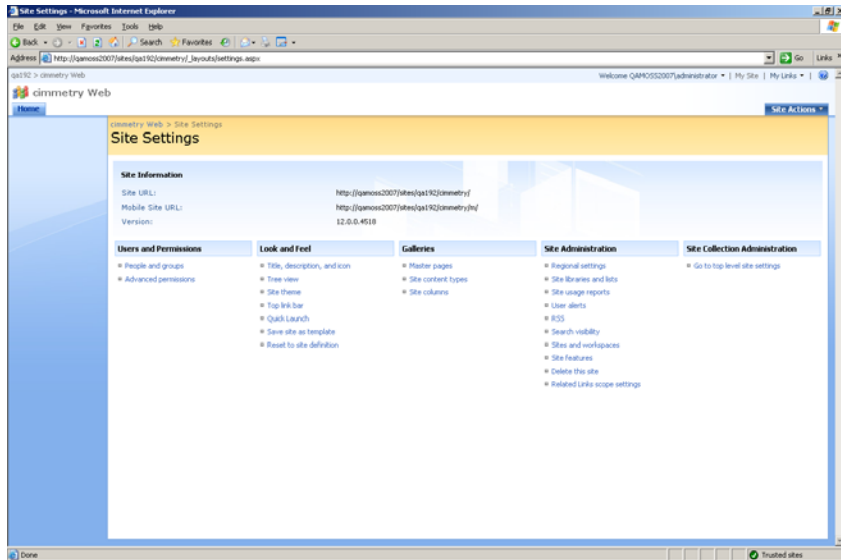


Note You need to manually update access and permission to the Cimmetry subsite for users and groups in your site. This is particularly important if a user wants to save and load markups and metafiles. The URL to the Cimmetry subsite administration page is on the previous page and the Installer will open it in separate browser once you click **Next**.

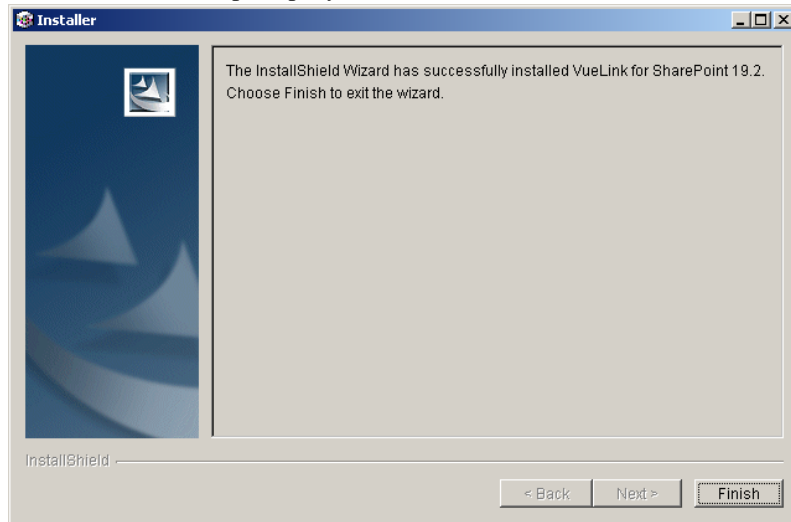
If you click the link for **cimSiteLog.out** you will see a page similar to the following:



- 15 If you choose to install the Windows SharePoint Services Components, the Cimmetry tool creates and verifies the Cimmetry subsite successfully. The Installer opens the Cimmetry Administration page for managing access to Cimmetry subsite. Follow the SharePoint documentation on how to manage anonymous access, users, and site groups for sites.



- 16 When the installation succeeds, the Installer shows the following information and prompts you to finish the installation:



- 17 Click **Finish** to quit the Installer.

Note When installation is complete, it is good practice to verify that the default values of XRef-related parameters inside vuelink.properties are valid (e.g. XrefCachDir path exists and matches the directory defined inside VueServer.ini in the AutoVue folder). For more information refer to [“Configuration and Support for XRef”](#).

Manual Installation

This section describes the steps necessary to install VueLink for SharePoint manually. In order for AutoVue Web Edition and SharePoint to work properly, you must perform the following steps.

SharePoint Customization

- 1 Verify that Microsoft IIS, ASP.NET and SharePoint (WSS or SPS) are installed properly.
- 2 Copy the **vue.aspx** file found on the distribution CD under /AutoVue_Web_Edition/dms_customization/TEMPLATE/LAYOUTS/1033 / directory to <SharePoint Root>/12/TEMPLATE/LAYOUTS/1033, where <SharePoint Root> is the root folder where SharePoint is installed (e.g. C:\Program Files\Common Files\Microsoft Shared\web server extensions).
- 3 Run the Cimmetry tool found on the distribution CD in the /AutoVue_Web_Edition/dms_customization/console / directory, from command line and specify the URL of Top Level Site under which Cimmetry site/lists have to be created. The Cimmetry site is used to store all markups and metafiles generated for any document within the Top Level Site hierarchy.
- 4 Edit **core.js** as described in [“Manual Steps for Customizing core.js”](#).

After running the tool a similar output should appear on screen:

```
Creating cimmetry elements in Site: http://mytoplevelsite
cimmetry Web created
  cimMarkup created
  cimMetafile created
  markupTable created
    BaseSiteURL created
    BaseListName created
    BaseFolder created
    BaseItemID created
    BaseVersion created
    MarkupSiteURL created
    MarkupListName created
    MarkupFolder created
    MarkupItemID created
    MarkupType created
    MarkupFileName created
  metafileTable created
    BaseSiteURL created
    BaseListName created
    BaseFolder created
    BaseItemID created
    BaseVersion created
    MetafileSiteURL created
    MetafileListName created
    MetafileFolder created
    MetafileItemID created
    RenditionType created
    MetafileFileName created
Cimmetry elements Created/Verified successfully
```

You can run this tool many times for a different or same Top Level Site. If the Cimmetry site or any part of it is already created, this tool only creates missing lists and fields. The following figure shows an example:

```

C:\Program Files\Cimmetry\UelinkForSharePoint\dms_customization\console>cimmetry
ysite http://qamoss2007/
Creating cimmetry elements in Site: http://qamoss2007/sites/guide1
cimmetry Web created
    cimMarkup created
    cimMetafile created
    markupTable created
        BaseSiteURL created
        BaseListName created
        BaseFolder created
        BaseItemID created
        BaseVersion created
        MarkupSiteURL created
        MarkupListName created
        MarkupFolder created
        MarkupItemID created
        MarkupType created
        MarkupFileName created
    metafileTable created
        BaseSiteURL created
        BaseListName created
        BaseFolder created
        BaseItemID created
        BaseVersion created
        MetafileSiteURL created
        MetafileListName created
        MetafileFolder created
        MetafileItemID created
        RenditionType created
        MetafileFileName created
Cimmetry elements Created/Verified successfully
C:\Program Files\Cimmetry\UelinkForSharePoint\dms_customization\console>

```

If the specified URL has extra information, this tool extracts TopSite URL from it automatically.

For example, if `http://hostname/sites/mysite/Shared Document/folder1` is passed as argument then `http://hostname/sites/mysite` is used as target automatically.

SharePoint Extension

- 1 Copy all Cimmetry web service files (asmx, aspx) found on the distribution CD under `/AutoVue_Web_Edition/dms_customization/ISAPI/` directory to `<SharePoint Root>/12/ISAPI/` where `<SharePoint Root>` is the root folder where SharePoint is installed (e.g. `C:\Program Files\Common Files\Microsoft Shared\web server extensions`).
- 2 Copy `CimWssExt.dll` from `/AutoVue_Web_Edition/dms_customization/ISAPI/BIN` directory to `<SharePoint Root>/12/ISAPI/BIN` where `<SharePoint Root>` is the root folder where SharePoint is installed.

VueLink Servlet and Web Application

VueLink Servlet files are found on the distribution CD under the /AutoVue_Web_Edition/vuelink_servlet directory.

The VueLink Web Application that is ready to deploy can be found on the distribution CD under the /AutoVue_Web_Edition/vuelink4j2ee directory.

The recommended approach is to copy the .war file from vuelink4j2ee directory into the deployment directory of the web application server. Note that you must restart the server if it does not support hot deployment. Otherwise, the server explodes the .war file into a directory with that name.

If you want to deploy the exploded version of the VueLink web application, create a folder with the same name as the .war file (e.g. vuelink in the case of vuelink.war) inside the application server deployment directory and copy all the content of vuelink4j2ee directory into it except for the .war file. You can also unzip the .war file into the new directory instead.

Note All applet launch configurations are stored inside the vuelink.properties file and are used to initialize the AutoVue applet. If you want to change the setting, you need to update this file. It can be found under WEB-INF\lib of the VueLink application.

Configuration

See [“Configuration”](#) for instructions for setting up VueLink.

Note that you must complete the steps outlined under [“Setting AutoVue Applet Parameters in vuelink.properties”](#).

Verification

See [“Verification”](#) for verifying that VueLink works properly.

Configuration

Setting DMS Launch Parameter in vue.aspx

- 1 Open the **Vue.aspx** file you copied under /12/TEMPLATE/LAYOUTS/1033 in step 2 of Installation DMS Extension, in a text editor such as Notepad.
- 2 Search for **__APPSERVER_HOST_PORT_CONTEXT__** and replace it with the **host:port/context** name of the VueLink application inside the web application server; note that “host” is the server name where VueLink is installed, “port” is the J2EE Application Server, and “context” is the VueLink web application name. For example: “SharepointServer:8080/VueLink”.

Manual Steps for Customizing core.js

During the automatic installation of VueLink for SharePoint, some javascript code will be added to **core.js** in your SharePoint directory. If you are installing VueLink for SharePoint manually, you must add the javascript code to **core.js** using a text editor as follows:

- 1 In the variable declaration section add:

```
////<Cimmetry_Customization_1>  
var csi_currentItemID=null  
////</Cimmetry_Customization_1>
```

- 2 Add the following code after the `m=CMenu(currentItemD+"_menu")` line in the function `CreateMenuEx(ctx, container, e)`:

```
////<Cimmetry_Customization_2>  
Custom_AddAutoVueMenuItem(m,ctx);  
////</Cimmetry_Customization_2>
```

- 3 Add the following functions to **core.js** (preferably after the AddDocTransformSubMenu(m, ctx) function):

```

////<Cimmetry_Customization_3>
function Custom_AddAutoVueMenuItem(m, ctx) {
var ObjType = GetAttributeFromItemTable(itemTable,
"OType", "FSObjType");
if ((ctx.isVersions || ctx.listBaseType == 1) && ObjType
!= 1 ) {
var strImagePath = "";
var strAction = "Custom_AutoVueScript(ctx)";
CAMOpt(m, 'View using AutoVue', strAction , null);
CAMSep(m);
}
return false;
}

function Custom_AutoVueScript(ctx) {
var httproot = ctx.HttpRoot;
var idx = httproot.indexOf("/");
if (!ctx.isVersions) {
csi_currentItemID = "Host=" + httproot.substring(idx + 2)
+ "&ListID=" + ctx.listName + "&ItemID=" +
currentItemID;
} else {
Querystring();
csi_currentItemID = "Host=" + httproot.substring(idx + 2)
+ "&ListID=" + Querystring_get("list",ctx.listName)
+ "&ItemID=" + Querystring_get("ID","");
var verURL = itemTable.verUrl;
ver = null;
var idx2 = verURL.indexOf("VersionNo");
if (idx2 > -1) {
var idx3 = verURL.indexOf("&",idx2+10);
ver = verURL.substring(idx2+10,idx3);
}
if (ver != null ) {
csi_currentItemID += "&Version=V" + ver+".0";
csi_currentItemID += "&VersionNo=" + ver;
}
}
window.open(ctx.HttpRoot+"/_layouts/" + L_Language_Text
+ "/vue.aspx?" +
csi_currentItemID,"AutoVue","location=1,resizable=1"); }

```

```
function Querystring() {
var
querystring=location.search.substring(1,location.search.length);
var args = querystring.split('&');
  for (var i=0;i<args.length;i++) {
var pair = args[i].split('=');
temp = unescape(pair[0]).split('+');
name = temp.join(' ');
temp = unescape(pair[1]).split('+');
value = temp.join(' ');
this[name]=value;
}
this.get=Querystring_get;
}

function Querystring_get(strKey,strDefault) {
var value=this[strKey];
if (value==null) {
value=strDefault;
}
return value;
}
////</Cimmetry_Customization_3>
```

Setting AutoVue Applet Parameters in vuelink.properties

- 1 Open the **vuelink.properties** configuration file that is inside WEB-INF\lib of the VueLink application.

Note You may need to extract the VueLink application and redeploy it. If you expect to change the vuelink.properties more than once, we recommend that you deploy the exploded version of the VueLink.

2 Modify the following settings:

```
jVueServer=localhost:5099
appServer=localhost:8080
appContext=_APPSERVERCONTEXT_
CodeBase=_CODEBASE_
```

Where

jVueServer	Is the host:port address of AutoVue server that VueLink application should communicate with. This value will be used to replace _JVUESERVER:JVUESERVERPORT_ inside csiApplet.html
appServer	Is the host:port address of web application server that is deploying VueLink application. This value will be used to replace _APPSERVER:ASPORT_ inside csiApplet.html
appContext	Is the context name of VueLink application inside web application server (e.g. /vuelink). This value will be used to replace _VUELINKCONTEXTPATH_ inside csiApplet.html
CodeBase	Is the path to jVue folder in web/application server This value will be used to replace _CODE_BASE_ inside csiApplet.html

Enabling Markup and Metafile Functionality for New Sites

If new topsites are created after VueLink has been installed, the SharePoint Administrator needs to create and configure the Cimmetry subsite so that you can save markups and metafiles. Take the following steps for the newly added topsites:

- 1 On the SharePoint server, run **Cimmetrysite.exe** **<URL_to_the_topsite>**.
Cimmetrysite.exe is included in the VueLink distribution and is located at **AutoVue_Web_Edition/dms_customization/console**.

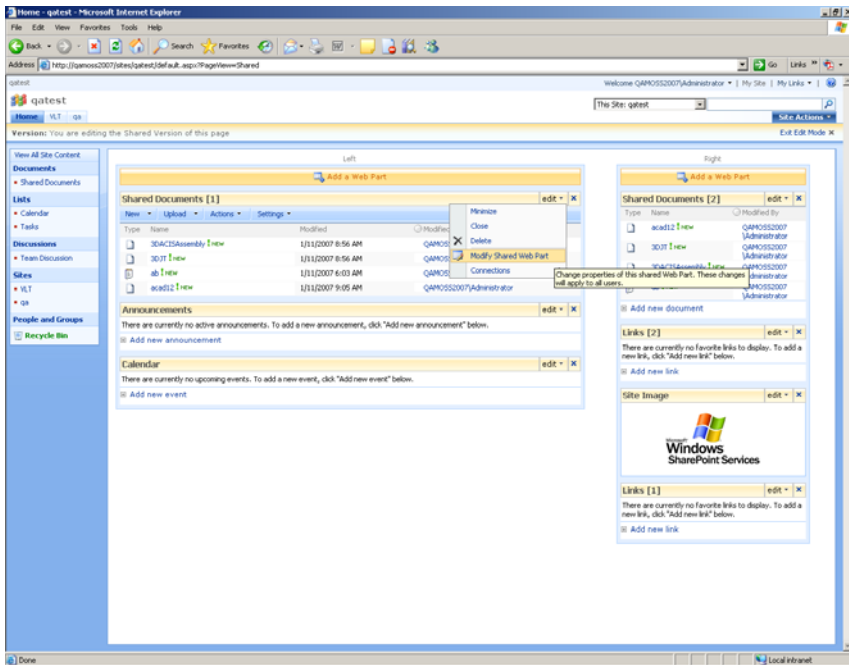
Running Cimmetrysite.exe creates a Cimmetry subsite under the specified topsite.

- 2 Logon to SharePoint as the Site Administrator and manually configure access and permission to the Cimmetry subsite.
Users with proper access can now save markups for files in the new topsite.

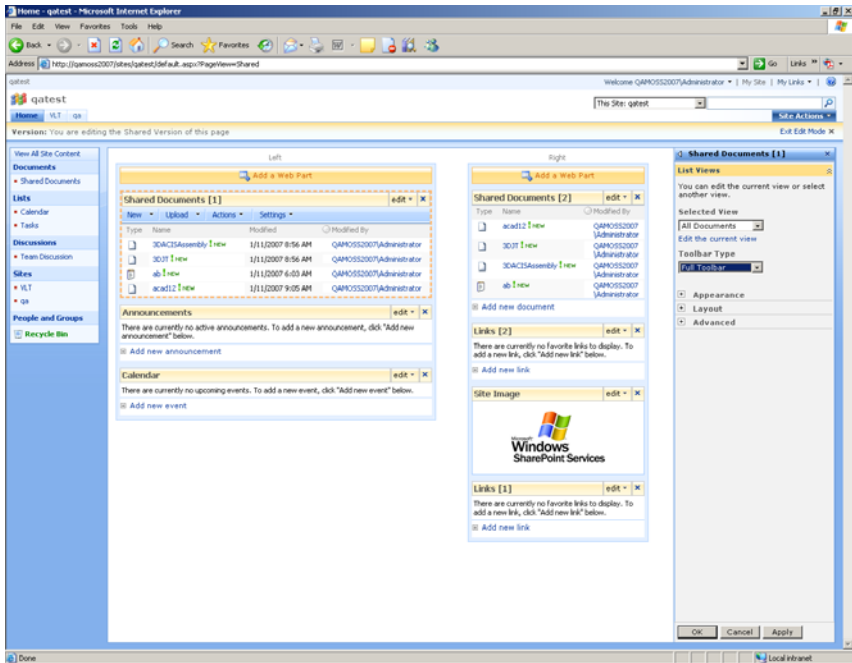
Enabling Viewing Files using Web Parts

Take the following steps to enable AutoVue viewing for files using Web Part:

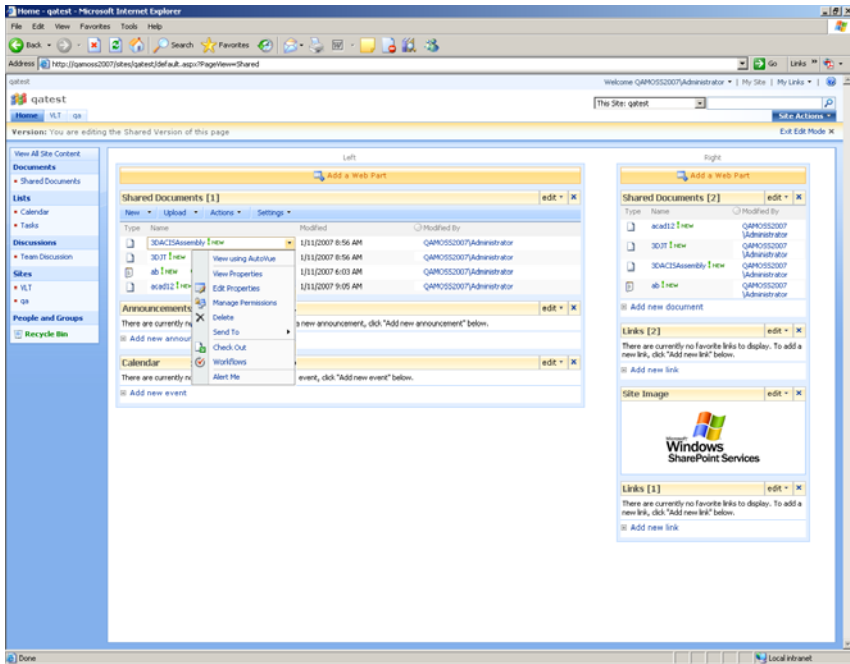
- 1 In Shared Documents, click **Edit > Modify Shared Web Part**.



- In the options that appear, select **All Documents** as the **Selected View** and **Full Toolbar** as the **Toolbar Type**, then click **Apply**.



Customized AutoVue options are now available.



Configuration and Support for XRef

VueLink for SharePoint has built-in support for External Reference Files (XRefs) that are declared inside a document.

Supported Mechanism

VueLink for SharePoint has support for two types of XRefs:

- Through check-in by CAD integration software packages like those from Organice™ and Bentley™.
- Through direct check-in from file system.

CAD Integration

VueLink for SharePoint supports SharePoint CAD integration such as Organice™ Explorer CAD Integration and Bentley™ ProjectWise. Please refer to documentation of these products for the XRefs attach/detach functionality. There are two ways that XRef paths can be saved inside a base document in the CAD Authoring application.

- If the XRef paths are absolute file system paths, you should check in the XRefs to the same place as base document inside SharePoint (same folder of same document library) since VueLink ignores the file system or UNC path for XRef resolution.
- If the XRef paths are relative paths to base document location, make sure the XRefs are checked in properly and relative to the location of base file inside SharePoint.

For example, the following shows how the XRefs path should be saved inside base file (if a base file exists in Folder1 of a SharePoint document library e.g. base file location: Site/List\Folder1\basefile.ext).

XRef location	Saved Path in Base Document (relative path)
Site\List\Folder1\xref1.ext	.\xref1.ext
Site\List\Folder1\subfolder1\xref2.ext	.\subfolder1\xref2.ext
Site\List\Folder2\xref3.ext	..\Folder2\xref3.ext
Site\List\Folder2\subFolder2\xref4.ext	..\Folder2\subFolder2\xref4.ext

File System

VueLink for SharePoint supports XRefs resolution for files that have been checked in directly from file system as long as the base file and its XRefs are properly saved into SharePoint, and that the XRef filenames have not been modified. Refer to [“Manual Steps for Enabling/Disabling XRef Support”](#).

Manual Steps for Enabling/Disabling XRef Support

VueLink for SharePoint XRefs support feature is customizable through some properties inside the **vueLink.properties** file. These properties come with predefined values to speed up the setup and configuration. The complete list of these properties and their valid values are listed below:

Note All of the property names inside **vueLink.properties** beginning with ‘Xref’ are related to the XRef functionality of VueLink for SharePoint and, unless it is being disabled, must have a valid value for XRef processing to work properly.

XrefBase	<p>Indicates the XRef resolution should be applied to what formats.</p> <ul style="list-style-type: none"> • To disable XRef functionality altogether, leave it empty (e.g. <i>XrefBase=</i>). • To apply Xref resolution to all formats, set this value to * (e.g. <i>XrefBase=*</i>). • For performance improvement, however, it is suggested to enable Xref resolution only for specific formats (e.g. CAD documents). This way no extra processing would be associated to the formats that have no Xref by nature (like most of the office type documents). • To enable Xref resolution for particular set of formats, use a colon (:) separated list of file extensions that are desired and the Xref processing should be applied to them (e.g. <i>XrefBase=dwg:dwf:prt</i>).
XrefValidExt	<p>Indicates the list of valid Xref file extensions.</p> <ul style="list-style-type: none"> • To disable Xref functionality, leave it empty. This way none of the Xrefs should be processed (e.g. <i>XrefValidExt=</i>). • If all Xrefs are valid and need to be downloaded along with the base file, set this property to * (e.g. <i>XrefValidExt=*</i>). • A colon (:) separated list means only Xrefs with defined extensions should be downloaded (e.g. <i>XrefValidExt=dwg:dwf:prt</i>).

XrefCacheDir	<p>Indicates the location inside the file systems to be used by VueLink for downloading and caching base files for Xref name extraction process.</p> <ul style="list-style-type: none"> An equivalent parameter should be defined inside AutoVue server option file VueServer.ini that maps to same location, so that AutoVue client can see the file for Xref name extraction process. The option inside VueServer.ini should be called Directory1 and set under the [Server] section. For example, if inside VueLink.properties there is: <i>XrefCacheDir=c:\\temp</i> Then inside VueServer.ini there should be: <i>[Server]</i> <i>Directory1= C:\\temp</i> <p>Note The configuration on AutoVue VueServer.ini file is only needed if server protocol is being used. For upload protocol it is not required. See XrefJVueProtocol property.</p> <p>Note Make sure the defined directory exists in the file system and VueLink and AutoVue have read/write permissions to this location.</p>
XrefJVueProtocol	<p>Defines the protocol used by VueLink to transfer file to AutoVue Server for Xref extraction process.</p> <ul style="list-style-type: none"> The default value is based on server protocol with Directory1 (e.g. <i>XrefJVueProtocol=server://@1/</i>). If you want to use different Directory number on the AutoVue Server, make sure to change @1 to a proper value. Alternatively, you can use upload protocol (which is useful if the AutoVue Server is running on a different server than VueLink host). In this case, you need to put the complete path to the location of VueLink Cache in the upload protocol (e.g. if <i>XrefCacheDir=c:\\temp</i> then <i>XrefJVueProtocol=upload://c:\\temp\\</i>).
XrefJVueServer	<p>Defines the connection used by VueLink to talk to AutoVue Server.</p> <ul style="list-style-type: none"> It can be either through socket or VueServlet (e.g. <i>XrefJVueServer=socket://localhost:5099</i>).
XrefJVueUser	<p>Defines the username used by VueLink to talk to AutoVue.</p> <ul style="list-style-type: none"> The default value is set to XREFUSER and there is no need to change it (e.g. <i>XrefJVueUser=XREFUSER</i>). <p>Note This would consume one seat of your license for the time that Xref processing is happening.</p>

Verification

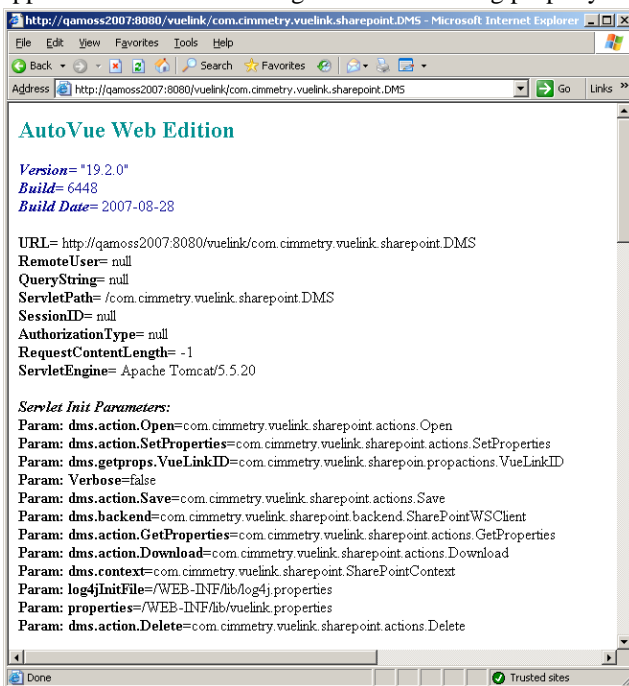
Verifying that VueLink is Running Properly

To verify that the VueLink Servlet (`com.cimmtery.vuelink.sharepoint.DMS`) is running properly, launch your web browser and enter the URL pointing to the Servlet alias name, which you assigned when installing `Vuelink.war` into the application server.

Here is an example of a URL:

`http://127.0.0.1:8080/vuelink/com.cimmtery.vuelink.sharepoint.DMS`

The following figure shows a sample response if VueLink is running properly. If you do not get a response similar to the one shown in the following figure, verify that the VueLink Servlet is installed and deployed properly and that your application server is running and is functioning properly.



The Build number and Build Date are also shown in the response above.

Running VueLink in Debug Mode

VueLink for SharePoint uses apache log4j package for logging. Default configurations are set in log4j.properties inside WEB-INF\lib folder of the VueLink application.

You can change the level and location of output by modifying this file. The following figure shows the different levels of logging available:

Will Output Messages Of Level					
	DEBUG	INFO	WARN	ERROR	FATAL
DEBUG					
INFO					
WARN					
ERROR					
FATAL					
ALL					
OFF					

For example, if you want to elevate log to DEBUG level then set **log4j.logger.com.cimmetry.vuelink=DEBUG** inside the previously-mentioned file.

For more information on log4j capabilities, refer to log4j documentation.

Feedback

Oracle AutoVue products are designed according to your needs. We would appreciate your feedback, comments and suggestions. We can be contacted by fax, e-mail or telephone. A new addition to our web page is a feedback button that activates an easy-to-use feedback form. Let us know what you think!

General Inquiries

Telephone: +1 514-735-3219

Fax: (514) 735-6440

E-mail: info@cimmetry.com

Web Site: <http://www.cimmetry.com>

Sales Inquiries

Telephone: +1 514-735-3219 or 1-800-361-1904

Fax: (514) 735-6440

E-mail: sales@cimmetry.com

Customer Support

Telephone: +1 514-735-9941

Web Site: <http://www.cimmetry.com/support>

