



EUROPEAN MEDICINES AGENCY
SCIENCE MEDICINES HEALTH

19 August 2010

Information and Communications Technology

PIM Data Validation Engine v 2.1.1

User Installation Manual

7 Westferry Circus • Canary Wharf • London E14 4HB • United Kingdom

Telephone +44 (0)20 7418 8400 **Facsimile** +44 (0)20 7418 8409

E-mail info@ema.europa.eu **Website** www.ema.europa.eu

An agency of the European Union



Table of contents

1. Introduction	3
2. System Requirements.....	3
2.1. Software Requirements	3
2.2. Hardware Requirements.....	3
2.3. Tested Environments	3
3. First-Time Installation Procedure.....	4
3.1. Step 1 – Access the EMEA URL.....	4
3.2. Step 2 – Download Application Installer.....	4
3.3. Step 3 – Automatic download and launch of new releases	4
3.4. Step 4 – Post-Configuration for PDF generation	6
3.5. Step 5 – Population of External Storage Area	7
4. Integration with third-party tools	8
4.1. Overview	8
4.2. Packages and libraries	8
4.3. Code examples	10
Classpath	10
Validate PIM Submission.....	10
Generate PDF format	11
5. Frequently Asked Questions	13
5.1. How to upgrade to the Latest JRE version (v1.6)	13
5.2. Find JRE version currently available on your computer.....	13
5.3. Install JRE 6.0 Update 3.....	14
5.4. Configure JRE 6.0.....	17
6. Document information.....	19
6.1. Open issues.....	19
6.2. Document location	19
6.3. Document Reference Id.....	19
6.4. Referenced documents.....	19
6.5. Document history.....	19
6.6. Review history.....	19
6.7. Distribution history	20

1. Introduction

This document describes the steps to be followed in order to download and install the **PIM Data Validation Engine**.

The engine consists of a simple front-end Java module that handles the Graphical User Interface (GUI) and a back-end Java module that performs the actual PIM validation and PDF generation. The document also provides some examples on how third-party tools can integrate with the back-end validation module in Java.

2. System Requirements

2.1. Software Requirements

The following Software is necessary in order to run the PIM Data Validation Engine:

- Java Run Time Environment from Sun (version 1.5 or higher).
- Web browser to view the HTML version of the product information via the PIM Viewer (Internet explorer version 6 or higher recommended – see note below)
- RedHat Liberation fonts to generate the PDF format (see section 3.4. for details)
- PDF reader (e.g. Adobe Reader) to view the PDF format

Where none of the above is available on the computer, they can be downloaded from the internet. See the EMEA web site (<http://pim.emea.europa.eu/lat/index.html>) for the URLs from which these tools can be downloaded.

Note: The View operation of the PDVE automatically launches by default the application associated with the “xml” file types in the user machine. In order for the PIM Viewer to launch a browser should have been designated as the application to open xml files. Furthermore, due to a known limitation with the handling of entities by the FireFox’s underlying XSL engine, the PIM Viewer works properly only with the Internet Explorer browser.

2.2. Hardware Requirements

The PDVE application will start with **64 MBytes** but more may be needed depending on the complexity of the product authored. The upper limit is set to **1024 Mbytes** but can be changed (see below)

Tip: You may configure the startup and maximum memory requirements by editing the `run_pdve.bat` and `run_pdve.sh` files located under the application deployment directory:

```
java -Xms64m -Xmx1024m -jar pimdes_validation.jar
```

2.3. Tested Environments

The DES Validation Engine has been successfully installed and operated using the following settings:

- Windows 2000 & JRE 1.5
- Windows XP & JRE 1.6.0


The DES Validation Engine is designed to operate on any Operating System that can support Java Run Time Environment 1.5 or higher.

3. First-Time Installation Procedure

3.1. Step 1 – Access the EMEA URL

Access the published URL for the DES Validation Engine from the EMEA PIM web site.

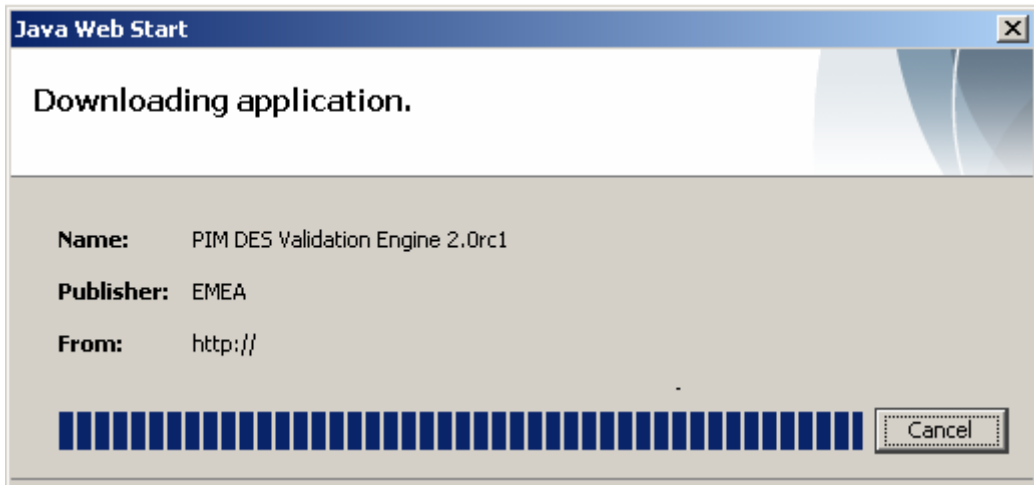
To download the latest version of the PDVE application, click on the icon below



Note
If Java is not launched automatically, you will need to install it on your PC!
Get it [here](#).

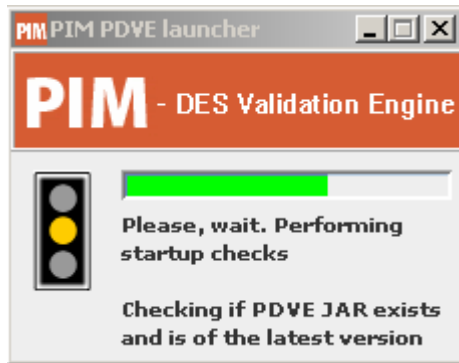
3.2. Step 2 – Download Application Installer

By clicking on the download link, Java Web Start will automatically launch and start downloading a custom application installer. During the download a window will display the download progress status:

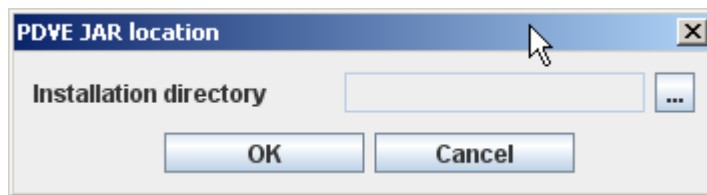


3.3. Step 3 – Automatic download and launch of new releases

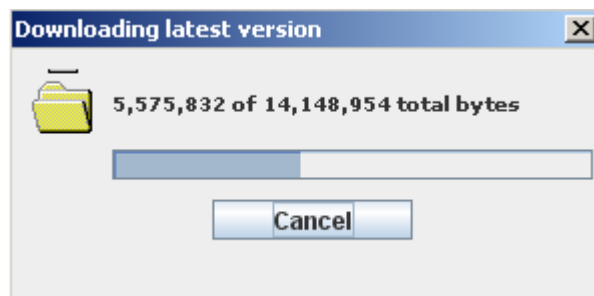
Once the custom application installer is downloaded, the following screen appears showing a progress bar for startup checks (“PIM PDVE Launcher”).



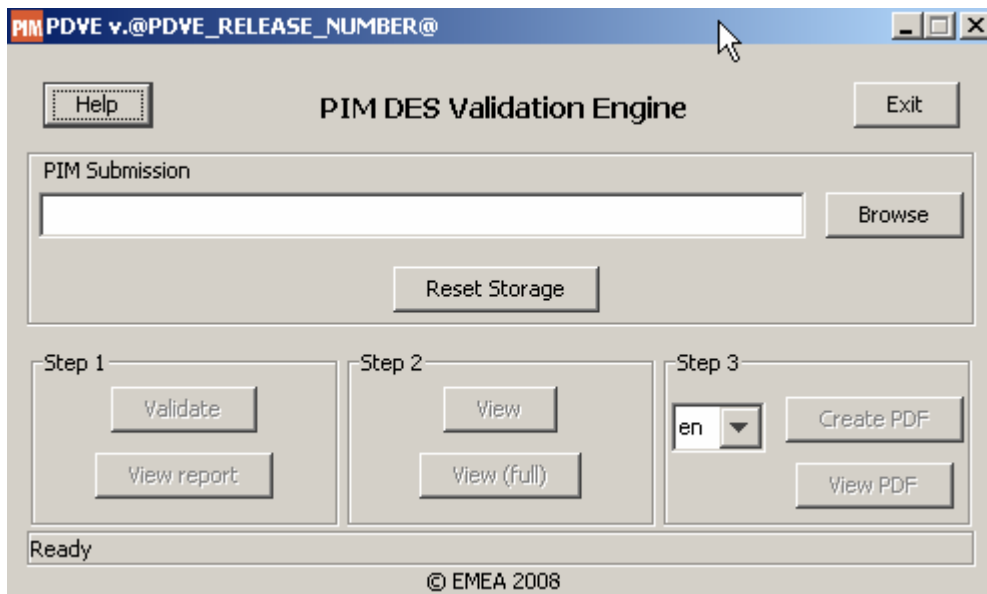
When downloading the application for the **first time** the user will be prompted to also specify an **installation directory**.



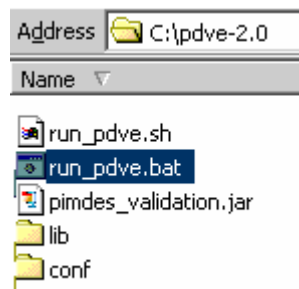
Next the installer will download the latest PDVE release from the EMEA website. The download time depends on the internet connection.



The latest release will be automatically downloaded and deployed under the designated installation directory and the application will automatically start up.

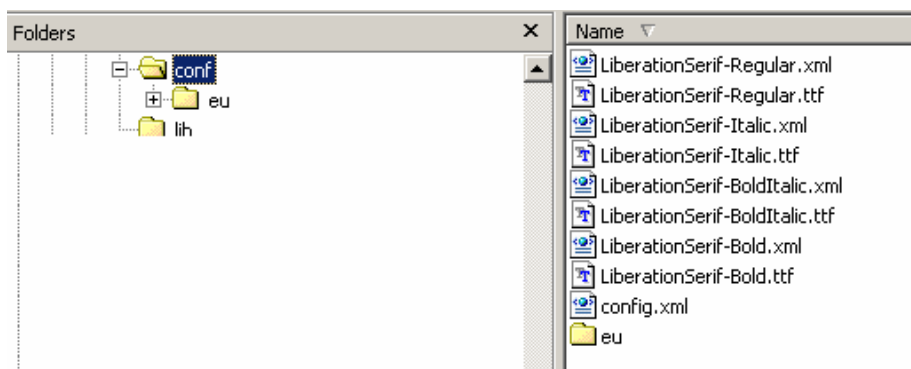


Important note: Each time the user attempts to start the application either from the **Java Web Start console** or from **the EMEA website download URL**, the system will automatically check whether a new version is available and automatically download it and deploy under the above specified installation directory. This automatic download and deployment can be avoided by launching the PDVE through some **batch** files instead. There are two **batch** files located under the application deployment directory, one for windows (**run_pdve.bat**) and one for Unix-like systems (**run_pdve.sh**):



3.4. Step 4 – Post-Configuration for PDF generation

Since **PDVE v2.1**, the **Liberation Serif** fonts required to generate the PDF format are included with the PDVE distribution. Once PDVE is installed, the fonts can be located under the PDVE **conf** directory. No special action is required by the user.



3.5. Step 5 – Population of External Storage Area

The PIM Data Validation Engine assumes that PIM submissions are placed under an external storage area (i.e. system folder) that the PDVE engine has both read and write operating system access: read access to retrieve the packages and write access to generate the various output formats (e.g. validation reports in html or pdf format, reconstructed pim.xml in case of delta submissions, complete product information documents in html or pdf format). For each product a single folder should be created containing all the PIM packages submitted under this product. The products folders do not have to be located under the PDVE deployment directory, as long as they are system accessible.

In the example below, a single folder has been created for the product-X and all PIM submissions (e.g. zip packages) have been manually copied by the applicant under this folder (one zip in this example).



Although the name of the folder does not have to be exactly the same as the one of the product, doing so will make maintenance of this repository easier (e.g. moving files around or navigating in the product directory structure).

Since PIM submissions may have references to previous ones, it is important that all valid product packages, applicant and regulator ones are located under this product folder. The same applies for packages submitted under different procedures. It is thus important that no specific folder is created per procedure.

4. Integration with third-party tools

4.1. Overview

This section of the installation manual describes how external applications can integrate with the back-end PDVE Java classes.

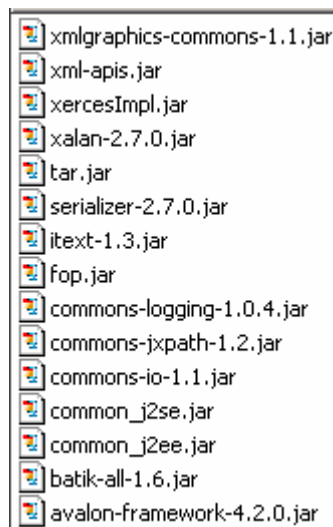
4.2. Packages and libraries

Once the installation process described in section 3.3 finishes, the designated **installation directory** will be updated as shown in the figure below. The highlighted files and folders are the important ones for integration with 3rd-party tools.

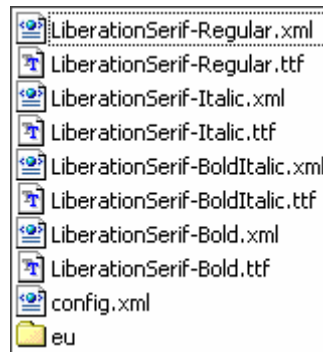


The **pimdes_validation.jar** contains both the front-end Java classes that handle the GUI for the stand-alone application as well as the back-end Java classes that implement the actual PDVE validation and PDF generation logic.

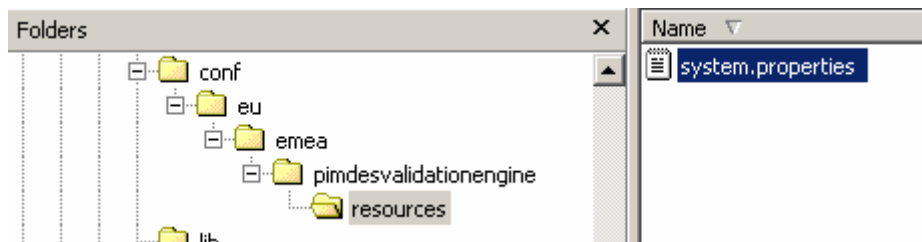
The **lib** folder contains all the external libraries (e.g. xml parser, fop engine etc.).



The **conf** folder contains a set of XML metric files created when the PDVE starts for the first time. Since PDVE v2.1, this folder also contains the actual liberation ttf fonts as described in **section 3.4** of this manual.



Important Note: As part of the integration process with another application the PDVE may need to be manually copied to a machine (e.g. server) other than the one used during the automatic download & deployment process. This can be done by copying the whole PDVE installation directory. Following this copy operation, the file `system.properties` located under the new `conf` directory in the target machine must be manually edited to reflect this transfer.



In particular, the last two properties `fop.base.url` and `fop.font.base.url` must point to the correct path in the new machine, as shown in the example below.

```
1 dtd.dir=eu/emea/pimdesvalidationengine/resources/
2 temp.file.debug.flag=false
3 xsl.root.dir=eu/emea/pimdesvalidationengine/resources/xslt/
4
5 # FOP-related stuff
6 fop.xsl.fc=eu/emea/pimdesvalidationengine/resources/xslt/converter/xhtml2fo5.x
7 xsl.root.context=eu/emea/pimdesvalidationengine/resources/xslt/
8 fop.xsl.config=eu/emea/pimdesvalidationengine/resources/xslt/conf/config.xml
9 fop.base.url=file:///C:/PDVE-RC-ATH/
10 fop.font.base.url=file:///C:/PDVE-RC-ATH/conf/
11
```

4.3. Code examples

Classpath

The `pimdes_validation.jar` can be used as an external library in 3rd-party tools to perform validation and PDFgeneration. Tools must include the JAR in their classpath in its installation directory.

More specifically, during compilation, the following entry must be present in the classpath

- `<installation directory>/pimdes_validation.jar`

During runtime, the following entries must be present in the classpath

- `<installation directory>/pimdes_validation.jar`
- `<installation directory>`

Validate PIM Submission

The following code example shows how to perform a PIM validation in a 3rd-party tool. Lines in bold indicate required code. Exception handling has been omitted for clarity. Some code lines have been prefixed with numbers for explanatory purposes.

```
import eu.emea.pimdesvalidationengine.context.StandaloneDesValidationContext;  
import eu.emea.pimdesvalidationengine.framework.base.BaseValidationReport;  
import eu.emea.pimdesvalidationengine.viewgen.ValidationReportGenerator;  
import eu.emea.pimdesvalidationengine.viewgen.beans.Report;  
import eu.emea.pimdesvalidationengine.client.Validation;  
...  
  
// create the context from the selected package  
1. StandaloneDesValidationContext context = new  
StandaloneDesValidationContext(pathToPackage);  
  
// execute validation  
2. BaseValidationReport report = Validation.validate(context);  
System.out.println("Report generated: " + report.toString());  
  
3. ValidationReportGenerator generator = new ValidationReportGenerator(context,  
report);  
  
// save HTML report  
4. Report htmlReport = generator.createHtmlReport();  
System.out.println("HTML report created: " + htmlReport.getFilename());  
  
// save PDF report  
5. Report pdfReport = generator.createPdfReport();  
System.out.println("PDF report created: " + pdfReport.getFilename());
```

Line 1 takes the full path to the target package (`pathToPackage` must have the absolute path to `<STORAGE-AREA>/<Product>/131-pim-XXXX-Y.zip`) and creates the fully reconstructed version in `<STORAGE-AREA>/<Product>/pdve/131-pim-XXXX-Y_full`.

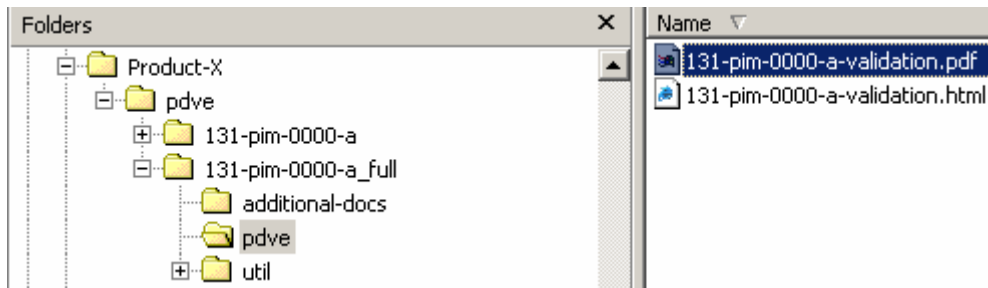
Line 2 performs the actual validation. The returned BaseValidationReport object may be used as is (e.g. custom results presentation). No information is saved in the file system at this point.

Line 3 initializes the report generator object for the creation of PDF and HTML reports.

Line 4 creates only the HTML version of the report in the file system (<STORAGE-AREA>/<Product>/pdve/131-pim-XXXX-Y_full/pdve/131-pim-XXXX-Y-validation.html). The returned object contains the HTML string as a UTF-8 byte array and the absolute path to the generated file.

Line 5 creates only the PDF version of the report in the file system (<STORAGE-AREA>/<Product>/pdve/131-pim-XXXX-Y_full/pdve/131-pim-XXXX-Y-validation.pdf). The returned object contains the PDF file as a byte array and the absolute path to the generated file. The HTML report is also silently generated in line 5.

The HTML and PDF formats of the validation report are generated under the folder <STORAGE-AREA>/<Product>/pdve/<131-pim-XXXX-a_full>/pdve so that they can be retrieved even without launching the PDVE application.



Generate PDF format

The following code example shows how to generate the PDF view(s) in a 3rd-party tool. Lines in bold indicate required code. Exception handling has been omitted for clarity. Some code lines have been prefixed with numbers for explanatory purposes.

```
import eu.emea.pimdesvalidationengine.context.StandaloneDesValidationContext;
import eu.emea.pimdesvalidationengine.viewgen.SubmissionViewGenerator;
import eu.emea.pimdesvalidationengine.viewgen.beans.SubmissionView;
...

// create the context from the package
1. StandaloneDesValidationContext context = new
StandaloneDesValidationContext(pathToPackage);

// prepare the generator with the selected language code
2. SubmissionViewGenerator generator = new SubmissionViewGenerator(context, langCode);

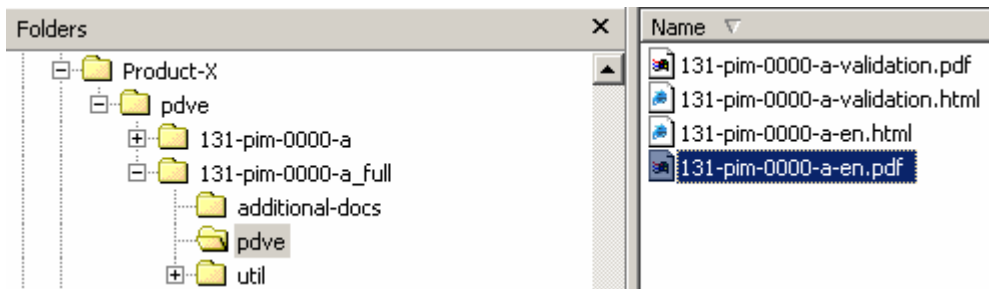
// return the list of generated views
3. List<SubmissionView> views = generator.createPdfViews(true);
for (SubmissionView view : views) {
    System.out.println("Generated view: " + view.getFilename());
}
```

Line 1 takes the full path to the target package (*pathToPackage* must have the absolute path to <STORAGE-AREA>/<Product>/131-pim-XXXX-Y.zip) and creates the fully reconstructed version in <STORAGE-AREA>/<Product>/pdve/131-pim-XXXX-Y_full.

Line 2 initializes the report generator object for the creation of the PDF view for the selected language. The value of *langCode* can be one of the values listed in interface *eu.emea.pimdesvalidationengine.constenum.LanguageCodeEnum*. All of the possible values result in the generation of a single PDF for the selected language. The only exception is *LanguageCodeEnum.ALL*, which causes the PDF generation for all the document's languages.

Line 3 generates the PDF view(s) and returns the list of objects to represent them. The boolean argument determines if the generated PDFs will be stored in the file system (*true*) or only returned as a byte array (*false*).

The HTML and PDF formats of the product information from all documents are generated, if the method argument is *true*, under the folder <STORAGE-AREA>/<Product>/pdve/<131-pim-XXXX-a_full>/pdve so that they can be retrieved even without launching the PDVE application.



5. Frequently Asked Questions

5.1. How to upgrade to the Latest JRE version (v1.6)

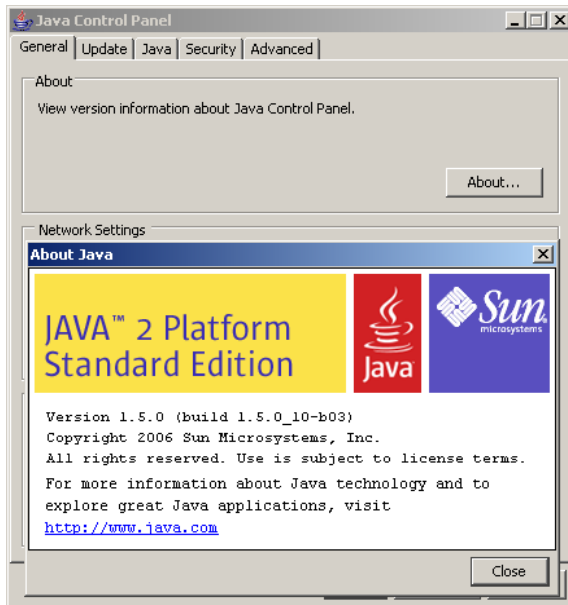
The DES Validation Engine can be used on a machine with a Java Runtime Environment (JRE) 1.5 or higher. In case the system is having an older JRE version (less than v1.5) or no version at all, the steps below should be followed:

Note: Since JRE 1.6 is the Latest version available in the Internet, the steps below illustrate how to upgrade/install this Latest release.

5.2. Find JRE version currently available on your computer.

Before you perform the following tasks, you should check that JRE 1.6 is not installed on your computer. To check this:

- Click on [Start – Settings – Control Panel – Java.](#)
- Click the [About button under the General](#) tab, and verify the version currently installed.

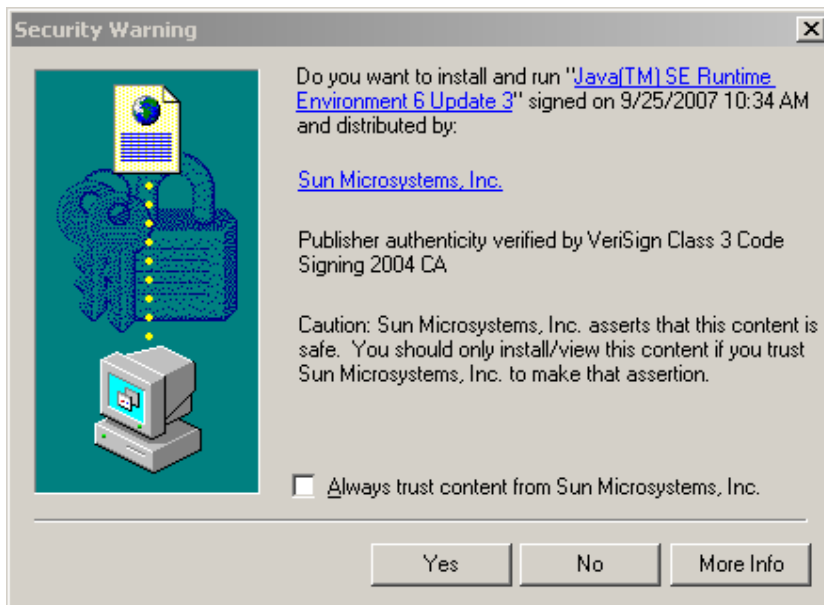


5.3. Install JRE 6.0 Update 3

1. Copy and paste the following link on your internet browser:
<http://www.java.com/en/download/index.jsp>
2. Click the [FREE JAVA DOWNLOAD](#) button.



3. Click Yes.



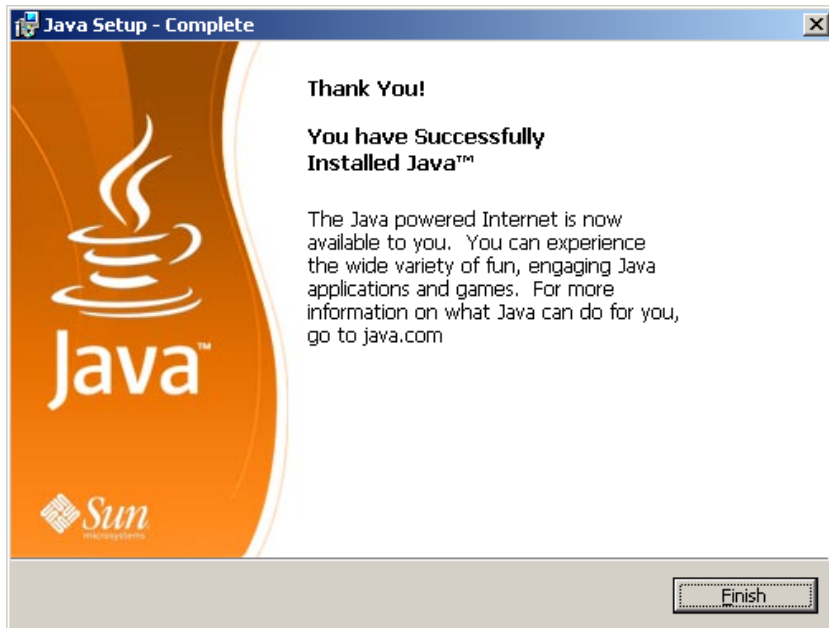
4. Accept the License Agreement.



5. Wait until all files are installed.



6. Click [Finish](#).

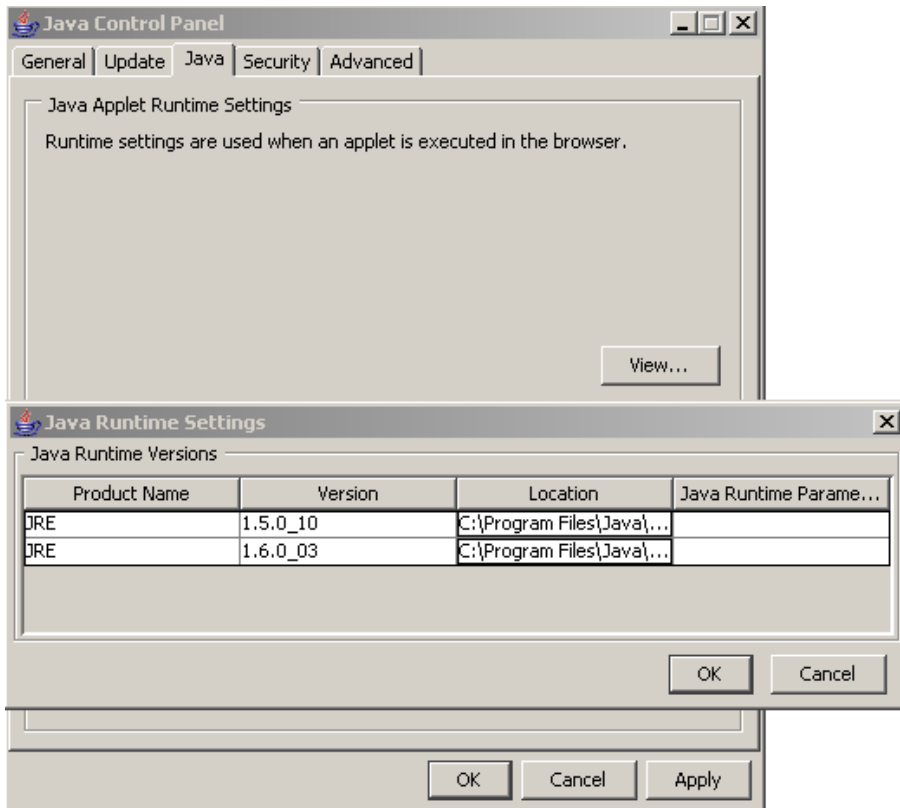


5.4. Configure JRE 6.0

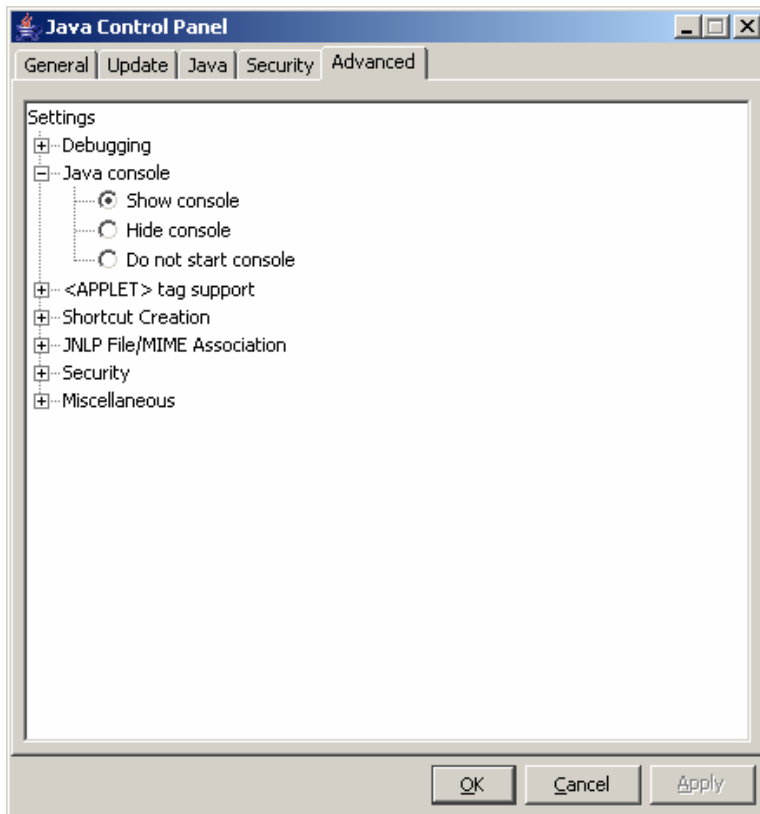
Note: At the time of writing this manual, the Latest update of JRE 1.6 was **JRE 1.6.0_03**.

Perform the following tasks to confirm that JRE 6.0 is installed on your computer and to enable the console.

1. Click [Start – Settings – Control Panel – Java](#).
2. Select the [Java](#) tab and press the [View](#) button. Verify that **JRE 1.6.0_03** is displayed as an available version.



3. Now close the front window and select the [Advanced](#) tab.
4. Select the option [Show console](#) and the press [OK](#). This is the option to display the Java console when the application runs, which will enable us to identify possible errors.



Notice: Every time JRE is installed, PDVE needs to be re-installed. However, the same deployment folder should be selected so the continuity is guaranteed.

6. Document information

6.1. Open issues

None

6.2. Document location

EMA

6.3. Document Reference Id

EMA/600545/2007

6.4. Referenced documents

6.5. Document history

Version	Who	When	What
0.1	K.Limitsios	17/12/2008	Initial Draft
0.2	K.Limitsios	23/12/2008	Note added on the need to associate a browser application with the "xml" type in order to launch the PIM Viewer from PDVE ("view").
0.3	K.Limitsios	20/08/2009	Updated for PDVE v2.1 (See section 3.4 - Fonts for PDF generation included with the distribution)
0.4	C.Humble	12/05/2010	Updated for PDVE v2.1.1

6.6. Review history

Version	Reviewer	When	Remarks
0.4	K.Limitsios	12/05/2010	

6.7. Distribution history

Version	Who	When