



MID
the modeling company

Innovator**Business**

Innovator**Object**

Innovator**Data**

Innovator**Function**

Innovator**Report**

Innovator Web

Understanding and Installing Innovator Web

Understanding and Installing Innovator Web–April 2010 Edition

Copyright © 1986-2010 MID GmbH Nuremberg, Germany. All rights reserved.

The copyright for Innovator software and the accompanying documentation is the property of MID GmbH.

Any reproduction or publication – in part or in whole – requires the written authorization of MID GmbH. Any violations of this will be prosecuted in accordance with copyright protection laws.

Registered trademarks and trade names are used in this documentation. Protection provisions apply to same, even if they are not marked as such.

The content of the documentation is for information purposes only and MID GmbH reserves the right to change it at any time without prior notice. MID GmbH will not be held liable for any damages arising from the use of the software or documentation.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

Contents

1	Foreword	1
2	Architecture of Innovator Web	3
3	Firewalls	7
3.1	Web Browser and Innovator Web.....	8
3.2	Innovator Web and License Servers.....	8
3.3	Repository Server and License Server.....	8
3.4	Innovator Web and Repository Servers	9
4	Technology Used	11
4.1	Servlets and JavaServer Pages	11
4.2	Tomcat	12
5	Runtime Environment	15
6	Installing Innovator Web	17
6.1	Preparing the Installation.....	17
6.2	The Installation Process	18
6.3	Enabling Model Log-In.....	18
	Creating User Groups	19
	Allowing Group Log-Ins for Innovator Web	19
	Logging-Out Logged-In Groups.....	20
	Other Information	20
6.4	Preparing Link to Documentation for Using Innovator Web	20
6.5	Troubleshooting.....	20
	Literature	23
	Index	25

1 Foreword

Innovator Web is a web application which runs as servlet¹ on any computer which offers the services of a servlet container. If any web browser connects with this computer, it is possible to view Innovator models of eXcellence or classiX editions which is the same as the view in the Innovator model browser as far as possible.

This guide describes Innovator Web architecture, requirements and installation.

The software and hardware names mentioned in this publication are registered trade marks and are therefore subject to legal regulations.

1. Java™ servlet technology offers a consistent mechanism to enhance the functionality of web servers and access databases. A servlet is comparable to an applet which runs on the server.

2 Architecture of Innovator Web

You can use Innovator Web to access Innovator models for reading (not modifying) from a common web browser (e. g. Mozilla, Opera, Netscape Navigator or Microsoft Internet Explorer).

As web browsers themselves are limited to certain protocols (HTTP¹ being the most popular), realization possibilities are limited.

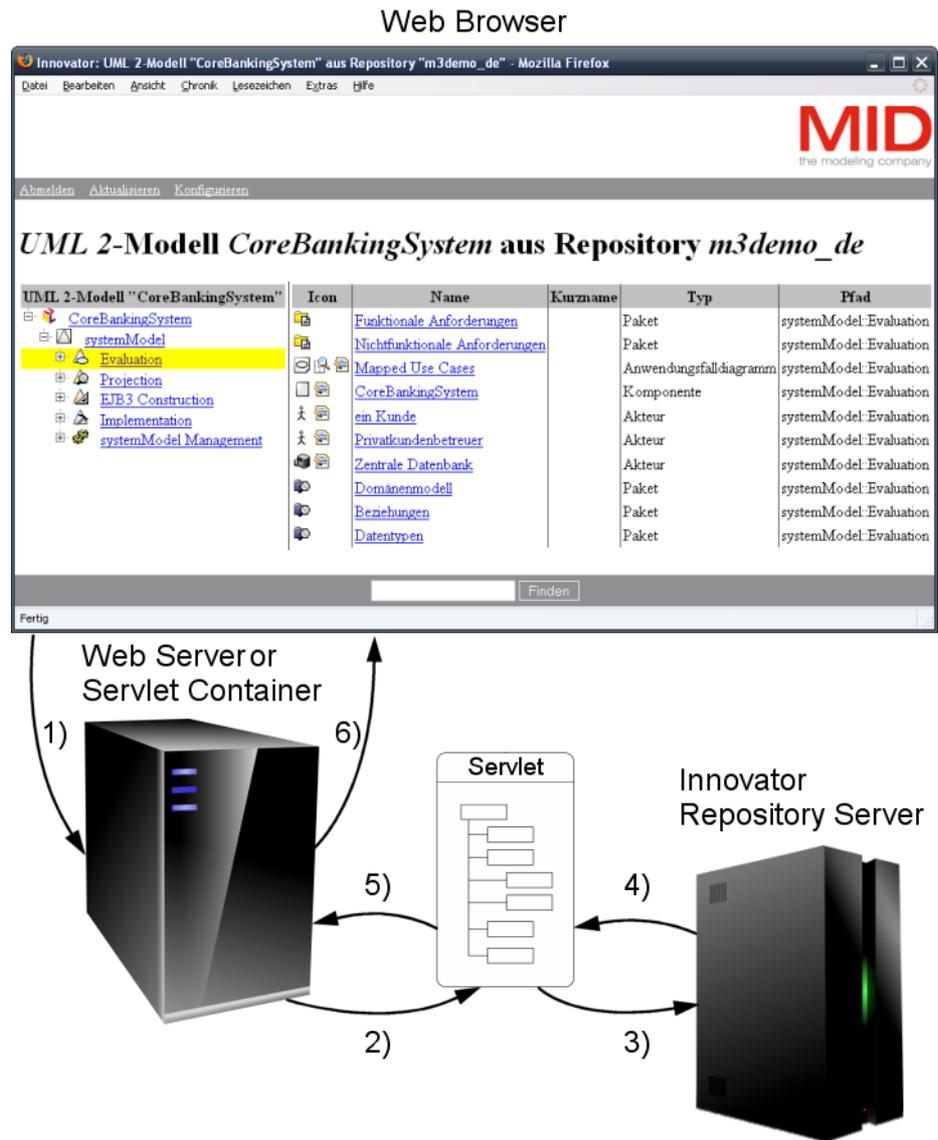
Generally, only two extensions are possible:

- Plug-ins, however, browser-specific plug-ins, and
- Java applets.

An architecture which realizes data transfer via HTTP or HTTPS² was chosen so that you are not limited in terms of using specific browsers or browser versions or by having to install a Java Runtime Environment.

Innovator Web is therefore a middleware which speaks HTTP, realizes web browser queries and transfers them to an Innovator server. Its answer then has to follow the same way back.

1. HyperText Transfer Protocol
2. HyperText Transfer Protocol Secure



The illustration above displays the resulting general architecture. This enables the web browser, web server and Innovator server to be run on different computers.

An Innovator Web query is therefore always carried out in this way:

- 1) Call the URL¹ in your browser (e. g. <https://foo.bar.de:8443/InoWeb2/>) which Innovator Web can be accessed from. Depending on the URL, the browser then sets up an unsafe ([http](http://) protocol) and/or a safe ([https](https://) protocol) connection.
- 2) The servlet container passes control to the servlet. The servlet has a so-called *session* for every enquiring web browser which it can memorize any number of states in. The user name and the objects that have just been displayed are normally among these states.

When accessing the servlet for the first time you need to enter a user group and password (log-in data) for the Innovator server to be used. The initial browser display is then shown. If one of the hyperlinks in this

1. Uniform Resource Locator, consists of transfer protocol, server address and port number if applicable

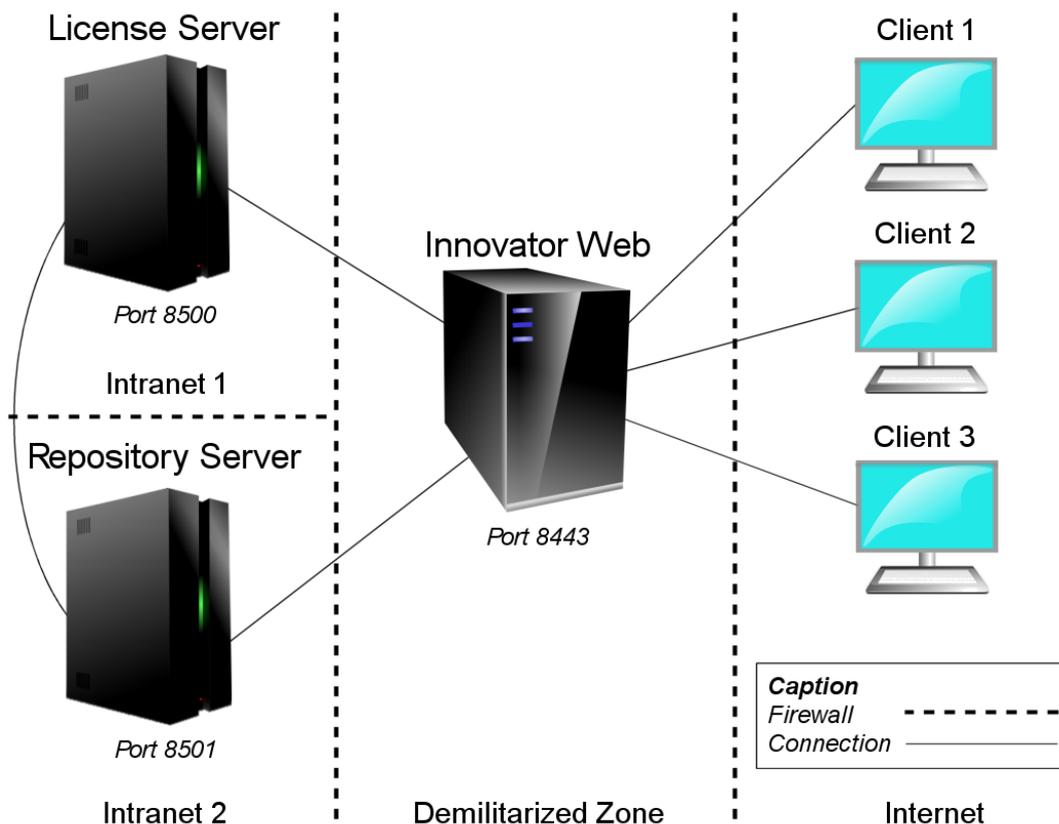
display is activated, the servlet is called again. Specific parameters mean that information is shown or the model's son packages are either shown or hidden.

- 3) The servlet now contacts the desired Innovator server and queries the necessary log-in data. When accessing the first time, this will be a log-in and, if the log-in is valid, the root node including its first package level. When accessing any other time, this will be the desired nodes or diagrams and/or tables as well.
- 4) The Innovator server sends the desired data to the servlet which completes the information saved during the session.
- 5) The servlet converts the updated object structure into an XHTML code and passes it to the servlet container.
- 6) The servlet container sends back this XHTML code to the browser via `http` and/or `https`; the browser then displays this code and provides hyperlinks for the objects. If you click on one of these hyperlinks, it then starts a new query.

You can choose between using a web server or a pure servlet container. For more information, see Chapter 5, "Runtime Environment", page 15.

3 Firewalls

As most companies working with Innovator are safety-conscious, it is essential to consider the interaction of Innovator, Innovator Web and firewalls.



The illustration above provides a good overview. The firewalls shown differentiate between intranet, demilitarized zone (DMZ) and internet. The intranet is divided into two zones by another firewall. Clients are located in the least reliable internet zone. Innovator Web runs in DMZ, the license server runs in the first intranet zone and the repository server runs in the second intranet zone

The ports which Innovator editions listen to connection requirements on are generally constant or at least predictable.

This also applies to Innovator Web. However, the ports where requirements for these ports are sent from are not predictable. This is not a problem within the scope of the firewall; on the boundary to the internet this

does not pose a problem either even though access to the Innovator Web server has to be admitted from all sides. This access is, however, limited to an arbitrary port for HTTP and/or HTTPS.

3.1 Web Browser and Innovator Web

The web browsers which perform the queries on Innovator Web can generally only communicate with the computer which Innovator Web runs on via HTTP and/or HTTPS¹.

You therefore only need to generally activate the appropriate port in the firewall or activate the respective clients if they have fixed IP addresses.

3.2 Innovator Web and License Servers

In order to operate Innovator Web, it is necessary to purchase an Innovator Web license and a corresponding number of Innovator Web client licenses. Innovator Web has to be able to establish a connection with the license server to be able to check this. This is done via a determined port specified by entering the fixed port in the INOHOST environment variable; this is only initiated from Innovator Web to the license server. i.e.: the license server only sends data if a query has previously been sent to Innovator Web; it does not send any data if no query has been sent. Therefore, in this example, Port 8500 has been activated for the server in DMZ which Innovator Web runs on.

3.3 Repository Server and License Server

The repository server contacts the license server in regular intervals. However, the license server may also call the repository server to determine whether it is still running (this behavior can be deactivated if necessary).

P is the port which is fixed for the license server. The repository server tries to connect to the p+1 port first. If this port is already assigned, the consecutive one is checked until a free port is found.

1. You can configure which of the two protocols should be used with which port in the servlet container and/or the web server.

This means that a firewall which might exist between the repository and the license servers has to allow all packages to pass to the license server's port –in reverse direction, all packages from the license server's port upwards (in practice however–depending on the number of running repository servers–the number will be limited to 20 - 50 ports).

3.4 Innovator Web and Repository Servers

Incoming connections from the Innovator Web server in DMZ to the ports are necessary for communication between Innovator Web and the repository server; the repository server can assign them.

Read Chapter 3.3, "Repository Server and License Server", page 8 for information about ports.

4 Technology Used

In order to make Innovator Web portable, it has been implemented as a *web application*. This is a collection of *servlets* and *JavaServer pages* (JSP) which can easily be installed. It then runs in a *servlet container*.

4.1 Servlets and JavaServer Pages

To be able to present dynamic web contents, it is necessary to have a mechanism in the web server which identifies data from specified sources following certain rules. Results are then calculated from these data, processed and sent back formatted in XHTML to the querying client. Vice versa, user entries should also be accepted.

Originally, only the *Common Gateway Interface* (CGI) could be used for this purpose. CGI scripts were and are often written in Shell syntax or Perl. One process each needs to be started with a Shell or with the Perl interpreter, the script needs to be loaded and then the process needs to be ended when running CGI scripts. As this is no high-performance process chain, procedures have been developed to reduce the efforts involved.

For *Apache* (certainly the most well known, flexible and safe web server) e. g. `mod_perl` is available. This is an extension which runs the CGI scripts in the same process as the web server itself by linking Perl to the web server.

Other extensions allow modules written in C or C++ to be linked to a web server. However, these are not easily portable and it is also quite hard to create them

The methods mentioned here are not suitable for creating complex business applications due to their disadvantages. This is one of the reasons why Sun developed Java Servlet Technology. Sun provides the following explanation about servlets:

"Servlets are the Java platform technology of choice for extending and enhancing Web servers. Servlets provide a component-based, platform-independent method for building Web-based applications, without the performance limitations of CGI programs. And unlike proprietary server extension mechanisms (such as the Netscape Server API or Apache modules), servlets are server- and platform-independent.

This leaves you free to select a "best of breed" strategy for your servers, platforms, and tools. Servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. Servlets can also access a library of HTTP-specific calls and receive all the benefits of the mature Java language, including portability, performance, reusability, and crash protection."

However, there are also some obstacles associated with the use of servlets despite all their advantages: XHTML code is embedded anywhere in the Java program code. The reason for this is that the servlets need to be able to generate XHTML. However, this is a rather annoying issue since any change made to the appearance of the web pages also means a change in the source code of the servlets. If several people are involved in creating a web application, greater problems quickly arise—especially when people with various skills are working together e. g. web designers and Java developers.

To react effectively, Sun has developed the JavaServer pages. Sun has made the following comments about this:

"JavaServer pages technology separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content.[...] JavaServer pages technology uses XML-like tags and scriptlets written in the Java programming language to encapsulate the logic that generates the content for the page. Additionally, the application logic can reside in server-based resources such as JavaBeans component architecture that the page accesses with these tags and scriptlets. Any and all formatting, HTML or XML, tags are passed directly back to the response page. By separating the page logic from its design and display and supporting a reusable component-based design, JSP technology makes it faster and easier than ever to build Web-based applications."

You need a JSP servlet specification and/or Java servlet specification implementation to be able to use JSP and, in doing so, the servlets.

4.2 Tomcat

Tomcat is the official reference implementation of the Java servlet technologies and JSP technologies. The current version is 4.1.x or 5.5.x and can be downloaded in binary or source code from <http://jakarta.apache.org/tomcat/>. It is developed within the "Apache Software Foundation" Jakarta project. This means that it is available within "Apache Software License" as *Open Source*.

The Apache Software License, Version 1.1

Copyright (c) 1999-2001 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgement: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>)."

Alternately, this acknowledgement may appear in the software itself, if and wherever such third-party acknowledgements normally appear.

4. The names "The Jakarta Project", "Struts", and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache" nor may "Apache" appear in their names without prior written permission of the Apache Group.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====
This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

5 Runtime Environment

You need a suitable runtime environment to run web applications; this consists of an installed Java SDK¹ and servlet container.

Generally, there are two kinds of servlet containers:

- **Standalone Servlet Container**
They do not need any additional web server and contain the code which is necessary for receiving and answering HTTP queries.
- **Embedded Servlet Container**
They can be embedded in a web server – via module mechanisms or other mechanisms. In this case, the web server receives the queries and passes them on to the servlet container where they are processed before the result is sent back.

Tomcat can either be used as standalone or embedded. The alternative is chosen depending on whether or not a web server is already running. Since use as an embedded servlet container depends on which web server is used, Innovator Web has been developed and tested with Tomcat 4.1.31 and 5.5.15 as a standalone solution. This does not, however, change anything about its behavior.

1. Software Development Kit - necessary for the simultaneous generation of Java code; can be downloaded from <http://java.sun.com/downloads.html> free of charge.

6 Installing Innovator Web

Certain prerequisites need to be fulfilled when installing Innovator Web. Read what these are and how to implement them in Chapter 6.1, "Preparing the Installation", page 17.

Read Chapter 6.2, "The Installation Process", page 18 for information about the installation itself.

The administrator needs to set up a group log-in in the model so that users can log-in to an Innovator model. For more information, read Chapter 6.3, "Enabling Model Log-In", page 18. You can find more detailed information about this topic in the Innovator user manual (keyword "Access Rights").

6.1 Preparing the Installation

Innovator Web accesses Innovator applications to display diagrams or tables. Therefore, Innovator 11 needs to be installed first before Innovator Web is ready for use. Beachten Sie beim Betriebssystem die Einschränkung auf bestimmte Versionen (siehe Datenblätter Innovator 11).

The user whose rights will later be used to execute Innovator Web has to add the path of the executable Innovator program files to their system search path (`%PATH%`).

System variables are also required; these define a corresponding environment for all users (see Administrator Manual, chapter 3.3, "Änderung der Installation unter Windows", page 30).

Load the license for operating Innovator Web and the client licenses in the administration program (`inoadm.exe`).

As well as Java 1.4.2 or newer versions, another prerequisite when installing Innovator Web is a functioning servlet container which—similar to Tomcat 4.1.31—implements the Servlet 2.3 and the JSP 1.2 specification and is also configured for the Java version mentioned above.

One of the servlets included in the servlet container should be executed to make sure it functions properly.

6.2 The Installation Process

Unpack the Innovator Web archive (`Innovator_Web_<version>.zip`) in the servlet container's directory intended for web applications.¹ This is `$CATALINA_HOME\webapps` for Tomcat.

Note If you already have an older version of Innovator Web installed in the `\webapps` directory, this old directory needs to be completely removed from the `\webapps` directory (not merely renamed); this is done so that faults do not occur upon application.

The Innovator Web archive (`Innovator_Web_<version>.zip`) includes the appropriate dynamically bound library.

Depending on the implementation applied, you might need to inform the servlet container where the necessary dynamically linked library which is needed to operate Innovator Web can be found.

Add the directory `%CATALINA_HOME%\webapps\InoWeb2\WEB-INF\classes\de\mid\api` to the `%PATH%` system search path under Windows.

Now start the servlet container. Innovator Web is now available.

Tomcat's default configuration now lists Innovator Web under `http://<hostname>:8080/InoWeb2`.

If Tomcat has been built with SSL support² or if an appropriate binary is available, you can assign a port for SSL connections using Tomcat by editing `$CATALINA_HOME\conf\server.conf` if required. Such a configuration already exists in the mentioned file but it is set on commentary. If you delete the comment and generate a key (instructions in this file, for more detailed information see Documentation), you will find Innovator Web under `https://<hostname>:8443/InoWeb2`.

6.3 Enabling Model Log-In

You can enable or block log-ins to Innovator models as group log-ins in the individual models for Innovator Web. You must be logged-in to the model as administrator to do this.

-
1. The web application can usually be introduced to the servlet container as a .war file. As Innovator Web uses the Java Native Interface to access a dynamically bound library, it is necessary to unpack the zip archive.
 2. SSL is the abbreviation for Secure Socket Layer

If you set up exactly one special user group (without password) for read access on Innovator Web, there is no log-in dialog when accessing the Innovator model.

Note

The following description applies to the Innovator classiX editions.

You can create new user groups in the model browser.

Creating User Groups

How to proceed

To create a new user group:

- » Jump to the model browser and log-in as model administrator.
- » Select [Model>Administration>Manage User...](#)
The dialog box of the same name appears.
- » Select the tab [Groups](#).
- » Enter the name of the new group in the field [Group list](#). Ensure that the name does not already exist in the list box [Group list](#).
- » Click on [\[New\]](#).
The new name appears in the Group List.
- » If necessary, assign a password using the [\[Password\]](#) button.
- » Close the dialog box.

You can allow or prevent group log-ins to the model via Innovator Web.

Allowing Group Log-Ins for Innovator Web

How to proceed

To allow group log-ins via Innovator Web:

- » Select [Model>Administration>Manage Logins...](#) in the model browser
The dialog box of the same name appears.
- » Jump to the tab [Groups](#).
The column [Logins](#) in the table shows how many group log-ins are available.
- » Click on the column [www](#) in the line which you want to change the group login in and select the desired entry.
These settings only apply if group log-ins are permitted.
- » Activate the check box [Allow group login](#).
- » Click on [\[Apply\]](#).
This enables group log-ins via Innovator Web.

Logging-Out Logged-In Groups

You can use Innovator Web to log-out groups which are logged-in.

How to proceed

To remove group log-ins via Innovator Web:

- » Select `Model>Administration>Manage Logins...` in the model browser
The dialog box of the same name appears.
- » Jump to the tab `Groups`.
- » Activate the check box `Logout all groups`.
- » Click on `[Apply]`.
The groups logged-in via Innovator Web are logged-out.

Other Information

- Innovator user manual, keyword "Access Rights" ("Transferring User Information to Another Model" etc.)

6.4 Preparing Link to Documentation for Using Innovator Web

The "Using Innovator Web" document is available as a PDF for Innovator Web users. As well as the link to the Innovator model, a link should be set for this PDF document in the browser so that users can access it.

6.5 Troubleshooting

Only Java error messages appear on the home page in the browser after installing and configuring Innovator Web? Some of the problems which cause these messages to appear can be easily fixed.

The default error message is:

```
java.lang.NoClassDefFoundError
de.mid.api.LicenseServer.<init>(Unknown Source)
```

There can be several reasons for this error message:

- Please check whether the license server is actually running.
- Please check whether you have access to the license server from the computer which the Tomcat server is running on. If there is a firewall, it needs to enable the port which the license server is running on.

If you have installed telnet, you can check whether there is a connection using "telnet <License Server Computer> <License Server Port>". If a window pops-up and then disappears again after a few seconds, the connection is OK.

-
- Please check whether the path component of the files being run ([PATH](#)) contains the path component of the current Innovator Executable ([INOEXE](#)) version.

Simple test: log-in with your login name that starts Tomcat, enter "inotext -v" in the command line and press the enter key. An inotext message window should open which contains the version recognition.

Check whether this version recognition is the current version!

Important: if Tomcat should run as a service under Windows, you need to restart the computer once the PATH has been set with the current INOEXE directory. Only then will the operating system adopt the path component.

If Tomcat runs as a program, it is enough to restart Tomcat.

- Only the current version of the InoWeb2 package can be installed in the [\webapps](#) Tomcat directory. All older versions need to be deleted from this folder. If you have to delete these, Tomcat then needs to be restarted.

Literature

- [1] Java Servlet Technology, <http://java.sun.com/products/servlet/>
- [2] Hans Bergsten, JavaServer Pages, 1. edition, O'Reilly, Cologne, 2001

Index

A

Architecture 4

B

Browser

-See web browser

C

Common Gateway Interface (CGI) 11

D

Demilitarized Zone (DMZ) 7

EEmbedded Servlet Container **15****F**

Firewall 7

H

HTTP 3

HTTPS 3

I

IN HOST 8

Innovator Web

-Installing 18

Internet 7

Intranet 7

J

JavaServer pages (JSP) 11

L

License server 7, 8

P

Port 7, 8

Protocol 3

-http 4

-https 4

Q

Queries

-General - run 4

R

Repository server 7, 8, 9

Runtime environment 15

SServlet **1**, 4, 11Servlet container 4, 11, **15**, 17

Session 4

Standalone Servlet Container **15****T**

Tomcat 12

U

URL 4

W

Web browser 3

MID GmbH

Headquarters Nuremberg
Eibacher Hauptstrasse 141
90451 Nuremberg
Germany

Tel.: +49 (0)911 96836-22
Fax: +49 (0)911 96836-10

E-Mail: info@mid.de
www.mid.de

Branch Cologne
Ettore-Bugatti-Strasse 6-14
51149 Cologne
Germany

Tel.: +49 (0)2203 20271-0
Fax: +49 (0)2203 20271-10

Branch Stuttgart
Silberburgstrasse 187
70178 Stuttgart
Germany

Tel.: +49 (0)711 633859-0
Fax: +49 (0)711 633859-10

Branch Munich
Keltenring 7
82041 Oberhaching
Germany

Tel.: +49 (0)89 95476831-0
Fax: +49 (0)89 95476831-9