



# ICW Master Patient Index 3.0

## System Administration Manual

Document version: 1.0 | Status: Approved

Security level: sensitive



## Imprint

InterComponentWare AG

Altrottstr. 31

69190 Walldorf, Germany

Tel.: +49 (6227) 385 – 0

Fax: +49 (6227) 385 – 199

E-Mail: [info@icw.de](mailto:info@icw.de)

© 2010 InterComponentWare AG. All rights reserved.

Document ID: 000c5187-8bee-2d10-bba9-df6bafb910d3

Document version: 1.0

Document language: en-US

Security level: sensitive

Product name: Master patient Index

Product release: Release 3.0

Last change: January 2011



## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
1.1	Document overview .....	4
<b>2</b>	<b>Brief Overview .....</b>	<b>5</b>
2.1	Starting and Stopping the System .....	5
2.1.1	Starting the system under Linux .....	5
2.1.2	Stopping the system under Linux .....	5
2.1.3	Restarting the system under Linux .....	5
2.1.4	Starting the system under Windows .....	5
2.1.5	Stopping the system under Windows .....	6
2.2	Directory Structure .....	6
2.3	Changing the database host .....	6
2.4	Changing the Directory Service (LDAP) .....	7
2.5	Checking Resource Use .....	7
2.6	Certifications .....	7
2.6.1	Storage of Server and Client Certificates in Apache2 .....	8
2.6.2	Storing server and client certificates in Apache Tomcat .....	8
<b>3</b>	<b>Managing the application .....</b>	<b>9</b>
3.1	JMX Management Console .....	9
3.1.1	Starting Java Management Console .....	9
3.2	Perform self-diagnostics .....	12
3.3	Basic Setting Options .....	13
3.3.1	HL7 Messages .....	13
3.3.1.1	Configure storage of corrupt HL7 messages .....	13



3.3.1.2	Deactivate storage of corrupt messages .....	15
3.3.1.3	Change directory for correctly imported HL7 messages .....	15
3.3.1.4	Deactivating storage of correctly imported HL7 messages .....	15
3.3.1.5	Changing directory for sent HL7 messages .....	16
3.3.1.6	Deactivating storage of sent HL7 messages .....	16
3.3.2	Validation settings for data entries .....	17
3.3.3	Setting the number of search results for a patient search .....	17
3.3.4	Setting the MSB URL for IHE PIX update notifications .....	17
3.3.5	Changing an automatic mapping in MPI .....	18
3.3.5.1	Activating automatic mapping and changing parameters .....	19
3.3.5.2	Mapping rules for scoring new datasets .....	20
3.3.5.3	Configure mapping of modified datasets .....	22
3.3.6	Automatic rescoring for insufficient data sets .....	23
3.3.7	Setting the sorting order for new matching tasks .....	24
3.3.8	Suppressing display of data sets .....	25
3.4	Advanced setting options .....	26
3.4.1	Changing the blocking settings .....	26
3.4.1.1	Specifying the algorithm .....	26
3.4.1.2	Recalculating the index .....	31
3.4.1.3	Replacement tables .....	31
3.4.2	Scoring settings .....	34
3.4.2.1	Changing the color coding for scoring values .....	34
3.4.2.2	Changing scoring parameters .....	35
3.4.2.3	Frequency tables .....	40
3.4.2.4	Source system-specific configuration .....	41
3.4.2.5	Recalculating scoring values .....	44



3.4.2.6	Repeat automatic mapping .....	44
3.4.3	Changing the filtering .....	44
3.4.4	Copying missing values .....	46
3.4.5	Changing the search expression converter .....	48
3.4.6	Automatically merge index patients .....	49
3.4.7	EHCache Configuration .....	50
3.5	Logging control .....	51
3.5.1	Setting the Log Level .....	52
3.5.2	Setting the Appender .....	52
3.6	Audit logs .....	53
3.7	Persistent modifications to log settings .....	54
3.8	Other log files .....	54
3.9	Bulk import of patient data .....	54
3.9.1	Configuring the file interface .....	55
<b>4</b>	<b>Configuring system properties .....</b>	<b>57</b>
4.1	System properties in the database .....	57
4.2	Tables .....	57
	<b>Glossary .....</b>	<b>65</b>
	<b>List of Figures .....</b>	<b>66</b>
	<b>List of Tables .....</b>	<b>67</b>



# 1 Introduction

This manual helps you to administer ICW Master Patient Index (MPI).

Several different system environments including system variables and files are addressed in this manual. The manual uses placeholders for this information, because they are dependent on your local system. You can find a list of placeholders in Table 11: Environment Variables and Table 12: Files and Folders.

Information about working with the application and about administration using the GUI can be found in the the User Manual and in the [Administrator's Manual](#).

## 1.1 Document overview

Section 1 contains information about this manual.

In section 2 you will find basic information about starting and stopping the application and about periodic system checks. Section 3 describes how to modify the configuration using the management interface at runtime. You will also find tips on logging.

More information about configuration settings can be found in section 4.



## 2 Brief Overview

### 2.1 Starting and Stopping the System

Start MPI both under Windows and Linux exclusively as a service. If you installed the MPI service as described in the Installation Manual, MPI will start automatically after your system has booted and will terminate as soon as you shut down the computer.



#### NOTE

Before you start the system, make sure that it is completely installed and configured. You can find information on installing and configuring the system in the Installation Manual.

#### 2.1.1 Starting the system under Linux

1. Log in as **root** user or as the respective **MPI user**.
2. in the command line, enter the following string:  

```
root@server:~> /etc/init.d/tomcatMpi start
```

#### 2.1.2 Stopping the system under Linux

1. Log in as **root** user or as the respective **MPI user**.
2. in the command line, enter the following string:  

```
root@server:~> /etc/init.d/tomcatMpi stop
```

#### 2.1.3 Restarting the system under Linux

1. Log in as **root** user or as the respective **MPI user**.
2. in the command line, enter the following string:  

```
root@server:~> /etc/init.d/tomcatMpi restart
```

#### 2.1.4 Starting the system under Windows

1. Open the Service Control Panel ([Control Panel] - [Administrator Tool] - [Service])
2. Double click the **tomcat-mpi** service
3. Click **Start**  
Alternatively, in the command line, enter:  

```
net start tomcat-mpi
```



## 2.1.5 Stopping the system under Windows

1. Open the Service Control Panel ([Control Panel] - [Administrator Tool] - [Service])
2. Double click the **tomcat-mpi** service
3. Click **Stop**

Alternatively, in the command line, enter:

```
net stop tomcat-mpi
```

## 2.2 Directory Structure

The following table provides an overview of some of the application-relevant folders. These folders may differ depending on your installation.

Directory	Description
<<CATALINA_HOME>> bin/ conf/	Executable files General configuration files
<<CATALINA_BASE>> conf/ logs/ webapps/ work/	Application-specific configuration files Log files, see <a href="#">Logging control [page 51]</a> section Packed and extracted web applications Session files
/etc/init.d/	Start scripts

Table 1: Directory Overview

## 2.3 Changing the database host

Modify the database properties as needed to suit your setup.

You can access the database properties via the file `server.xml`. This file is located in the `<<CATALINA_BASE>>/conf` directory.

```
...
<!-- Global JNDI resources -->
<GlobalNamingResources>
  <!-- Active config db -->
  <Environment name="dbUrl" type="java.lang.String"
value=" jdbc:oracle:thin:@<<db-host>>:1521:<<db-schema>>" />
  <Environment name="dbUsername" type="java.lang.String"
value="<<db-user>>" />
  <Environment name="dbPassword" type="java.lang.String"
value="<<db-password>>" />

```





```
<Environment name="dbDriverClass" type="java.lang.String"
value="<<db-driver>>" />
.....
```

## 2.4 Changing the Directory Service (LDAP)

LDAP (Lightweight Directory Access Protocol) is required for user management. Attributes for connecting to the LDAP are not configurable via JMX. Modify these attributes directly in the database table `property_configuration`, see Attributes of the Search Expression Converter, [Changing the Search Expression Converter \[page 48\]](#) section.

Modify the following values:

- `java.naming.provider.url` (URL for LDAP server)
- `java.naming.security.credentials` (password)
- `java.naming.security.principal` (user/contractor)

If you are using two replicating LDAP services in a high-availability environment, enter both URLs in `java.naming.provider.url`, separating them with an empty space, as in: `ldap://host1:389 ldap://host2:389`. If `host1` fails, users will be re-routed to `host2` and reverse.

## 2.5 Checking Resource Use

Check the availability of the most important application resources periodically using the control tool JConsole.



### NOTE

JConsole is a component of the Java JDK 1.6 distribution. In order to use JConsole, the JDK 1.6 must be installed on the administration workstation.

Monitor the following values:

- Free space in memory
- CPU (usage) and number of open threads
- Log files

## 2.6 Certifications

The certificate for operating MPI is stored in two different locations:

- In Apache2 on the Web server.
- In Apache Tomcat on the application server

## 2.6.1 Storage of Server and Client Certificates in Apache2

The following applies to a MPI installation under Linux.



### NOTE

Note: The mapping specifications used here refer to the web server upstream of the application. A standard Apache2 installation in the directory `/usr/local/apache2` is assumed.

If the web server is located in a different directory, `/usr/local/apache2` must be replaced by the applicable path.

1. Save the server certificate for the application to the following directory of the Apache2 server: `/usr/local/apache2/conf/ssl`
2. All client certificates must be stored in the following directory of the Apache2 server: `/usr/local/apache2/conf/ssl.crt`
3. Modify or expand the configuration file (`/usr/local/apache2/conf/httpd.conf`) of the httpd server.
4. Attach the information to the server and client certificates as shown in the example below:

...

```
# Server certificates
```

```
SSLCertificateFile /usr/local/apache2/conf/ssl/server.crt
```

```
SSLCertificateKeyFile /usr/local/apache2/conf/ssl/server.key
```

```
# Client certificates
```

```
SSLCACertificateFile /usr/local/apache2/conf/ssl.crt/cacert1.pem
```

```
SSLCACertificateFile /usr/local/apache2/conf/ssl.crt/cacert2.pem
```

5. Restart Apache2 once you have made your changes.

## 2.6.2 Storing server and client certificates in Apache Tomcat

1. Use the Java keytool to import all client certificates to the file `truststore.jks` in the directory `<<CATALINA_BASE>>/conf/`.
2. Use the Java keytool to import the server certificate to the file `keystore.p12` in the same directory (`<<CATALINA_BASE>>/conf/`).
3. Import the certificates.
4. Restart Apache Tomcat.

## 3 Managing the application

### 3.1 JMX Management Console

JConsole is a tool for monitoring the Java VM. It is also used to control suitable applications by the use of JMX (Java Management Extensions) compliant, so-called MBeans.

MPI provides for a number of configuration parameters over MBeans. Using a JMX client, such as JConsole, these parameters can be modified while running. The system does not need to be restarted. In addition, unless stated otherwise, MPI backs up all parameters exposed via MBeans in the database so that the previous valid values can be reused if the system is restarted.

JConsole is a component of the Java JDK 1.6 distribution. In order to use JConsole, the JDK 1.6 must be installed on the administrator workstation.

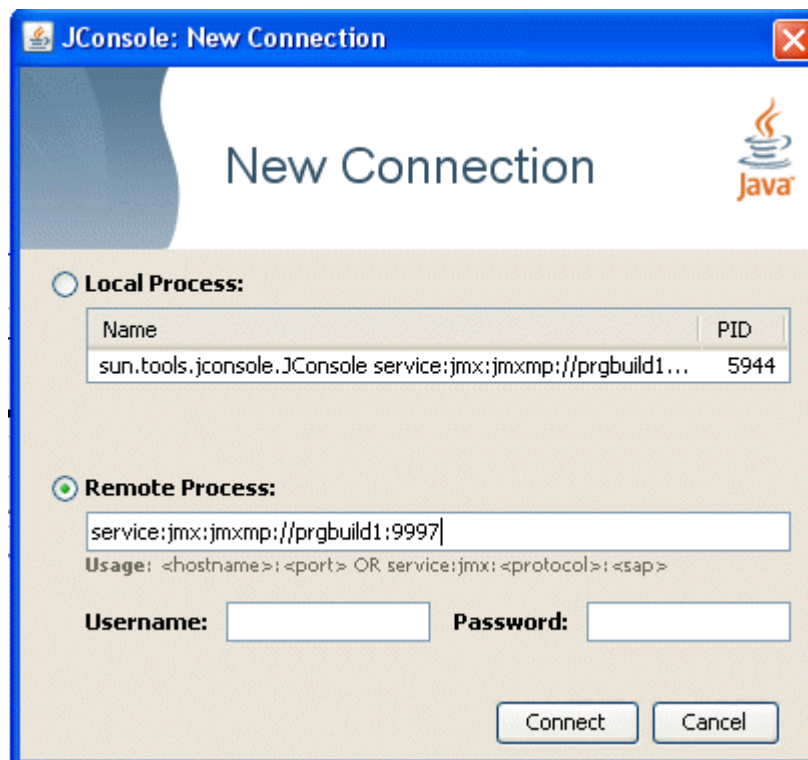


Figure 1: Login in JConsole

#### 3.1.1 Starting Java Management Console

If you are using JConsole for the first time, copy the included file `jmxremote_optional-1.0_01-ea.jar` from the Setup directory to the directory `jre/lib/ext/` under your Java home directory (`<<JAVA_HOME>>`).

1. Open JConsole in the `<<JAVA_HOME>>/bin` directory.
2. Select **Remote Process**.
3. Under **Remote Process**, enter the JMX URL:  
`service:jmx:jmxmp://<host>:<port>`.

The default port of the JMXMP protocol is 9997. The currently configured port is defined in the `property_configuration`, in the `jmx.jmxmp.service.port` item, see *Attributes of the Search Expression Generator* in the [Changing the Search Expression Generator \[page 49\]](#) section.

Examples for the JConsole interface screen can be seen in the two figures below.

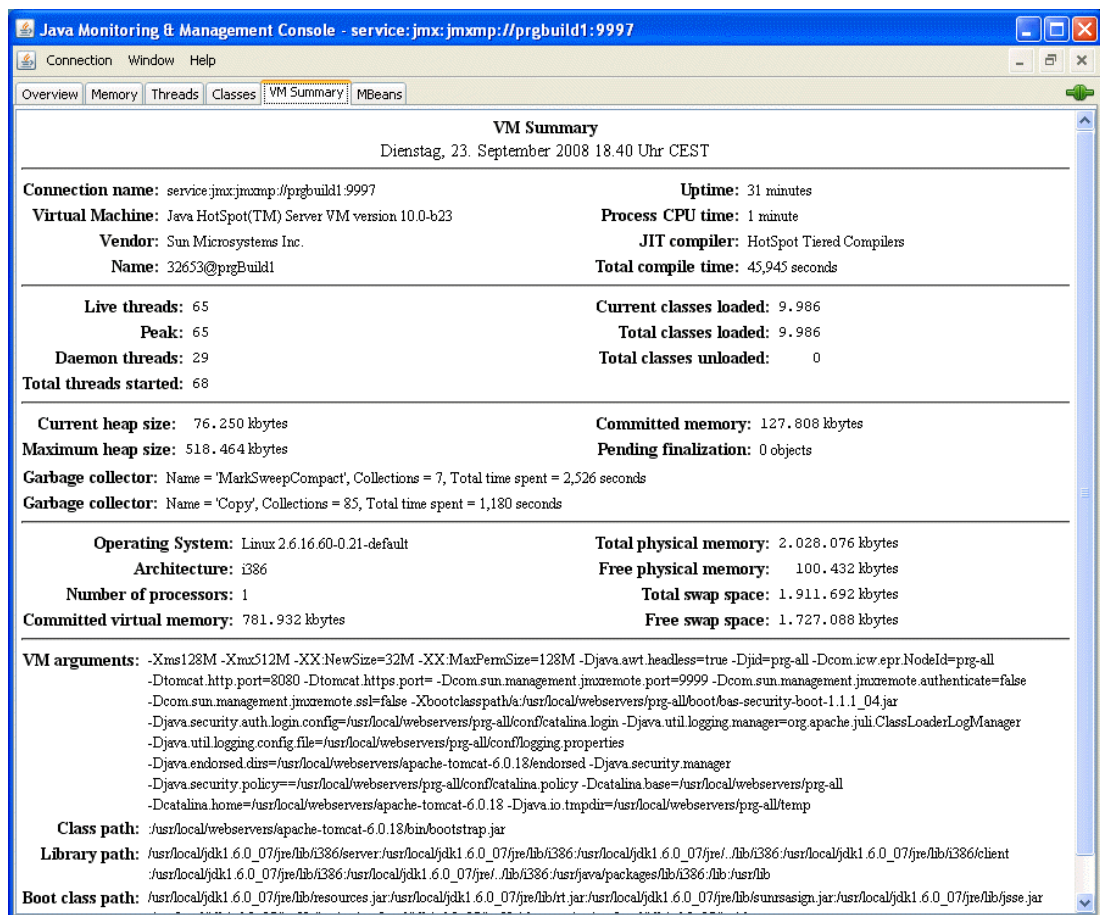
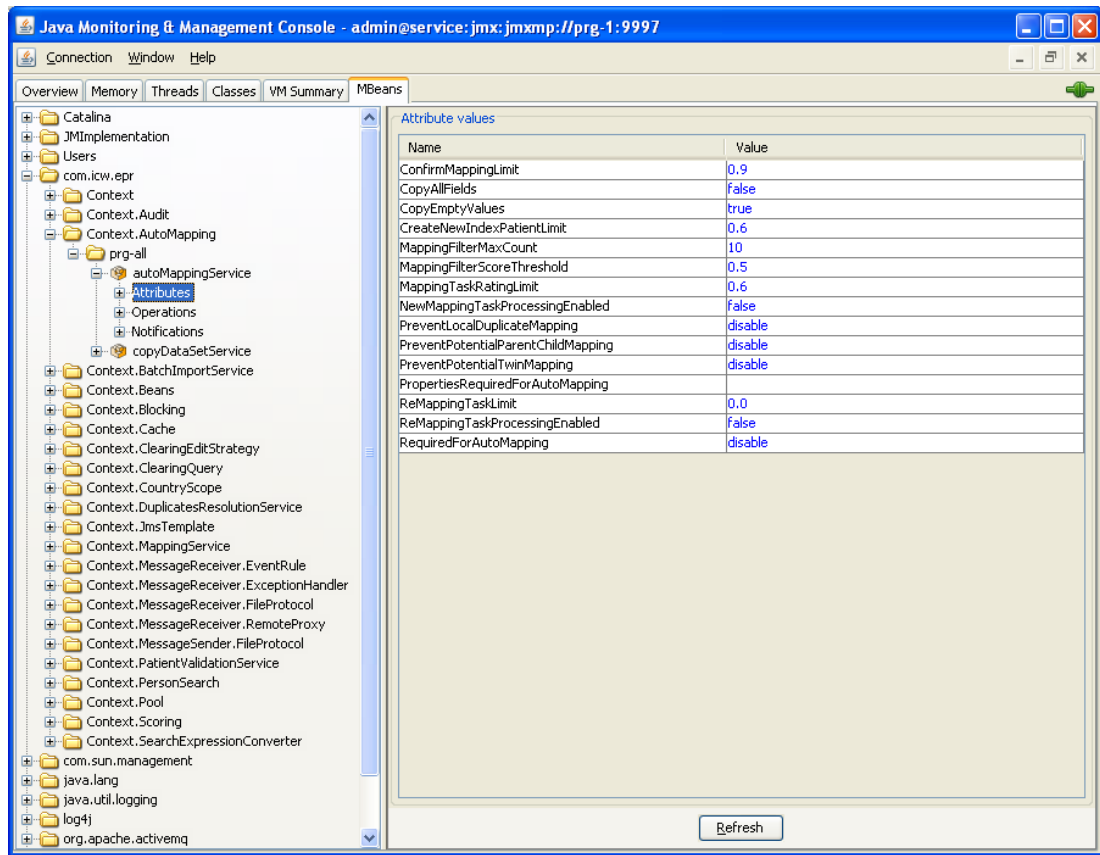


Figure 2: JConsole overview

The *MBeans* view shows all modifiable configuration parameters.



**Figure 3: JConsole MBeans**

The available MBeans are divided into the following groups:

MBean	Description
Catalina	Tomcat-specific configuration options, e.g. cache settings, connection settings, etc.
JMImplementation	Java Virtual Machine status information
Users	Tomcat user management
com.icw.epr	All of the application-specific configuration values shown in Table 9.
com.sun.management	Enables the execution of heap dumps over the HotSpot VM
java.lang	Java-specific configuration values
java.util.logging	Standard Java logging classes
log4j	Logging-specific configuration values
org.hibernate	Database-specific configuration values
prg-all-logging	Logging-specific configuration values

**Table 2: Summary of MBeans**



## 3.2 Perform self-diagnostics

1. Enter the following URL in the address line of your browser:

`http://<host>:<port>/mpi/common/Status.info`

- ⇒ MPI returns a short status page. No authentication is necessary. This status page is a text-only page and, when successful, follows the following format:

OK

ActiveMQ: [OK: [QueueSize: 0]]

Database: [OK: [Database working, Menu table accessible]]

LDAP: [OK: [LDAP working, User <system> found]]

This status page enables the monitoring tools to determine the status of MPI automatically:

- Line 1 contains either **OK** or **FAILED**. If **OK** the application should function without any limitations. In the case of **FAILED**, important application components are not available.
- Line 2 gives information about the availability of the internal message buffer and about the size of the matching tasks. A number greater than 0 does not necessarily mean a problem exists, but the size of the queue should drop steadily, especially when only a small number of incoming HL7 messages are being received. Continually high numbers indicate a low matching processing speed.
- Line 4 gives information about the availability of the database. This is determined by reading out the MPI menu table. If the database is not available it will not be possible to work with the application.
- Line 4 gives information about user database (LDAP) availability. If LDAP becomes unavailable users will no longer be able to log in and will not receive any new HL7.

### Status messages

You can also query the three status messages in lines 2 to 4 individually. To do so use one of the following URLs:

- `http://<host>:<port>/mpi/common/StatusActiveMQ.info`
- `http://<host>:<port>/mpi/common/StatusDatabase.info`
- `http://<host>:<port>/mpi/common/StatusOpenLDAP.info`

## 3.3 Basic Setting Options

### 3.3.1 HL7 Messages

MPI receives data via asynchronous importing of HL7 messages. Among other methods. HL7 messages include patient registrations, generation of encounters (administrative cases), and registration of documents. Additionally, the program itself generates HL7 messages when datasets are linked to one another in the Master Patient Index (MPI), or when a mapping is dissolved.

These messages can be logged in the file system. This helps you keep track of which data has been imported and exported by the system.



---

**CAUTION      Data privacy**

The messages often contain sensitive patient information, such as periods of hospitalization and diagnoses. For this reason ensure that only authorized persons have access to the configured destination folders in the file system.

---

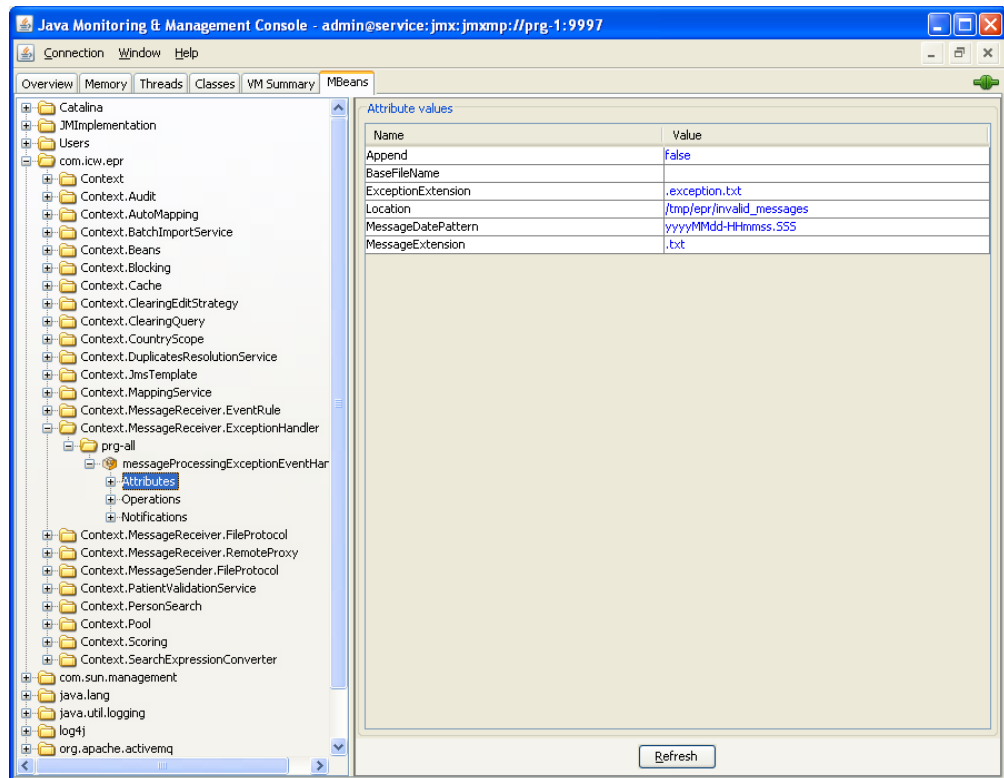
#### 3.3.1.1 Configure storage of corrupt HL7 messages

If MPI receives an HL7 message that cannot be processed due to errors in syntax or semantics, that message is saved to a directory provided specifically for that purpose. There, corrupt or incorrect messages can be stored for analysis at a later time.

1. Select the **MBeans** tab.
2. Select the MBean with the following names:  

```
com.icw.epr:type=Context.MessageReceiver.ExceptionHandler,  
Context=prg-all,  
name=messageProcessingExceptionHandler
```

⇒ The details view of the component will open, showing the selected MBean. The *Attributes* MBean is activated, see the next figure



**Figure 4: MBeans, corrupt HL7 messages**

3. In the *Location* field, enter the desired directory name on the server.



**NOTE**

If the directory does not yet exist the system will create it automatically.

⇒ Corrupt messages will be saved to this directory.

4. Fill out the fields with any other needed changes as indicated in the following table.



**NOTE**

Any text can be used as file extensions that would lead to a correct file in the file system when attached to an automatically generated name.

5. Press the Enter key after each change.

Field	Value
MessageExtension	File extension for corrupt messages
ExceptionExtension	File extension for the corresponding error message
MessageDatePattern	Describes the file name format, depends on the point in time the file was created
BaseFileName	Common portion of the file name



Field	Value
Append	Specifies whether a file will be created for each new message, or whether it will be appended to an existing file, unless otherwise affected by MessageDatePattern
Location	Specifies the directory where corrupt messages will be filed

**Table 3: Mbean: Saving corrupt HL7 messages**

### 3.3.1.2 Deactivate storage of corrupt messages

1. In the MBean, set

```
com.icw.epr:type=Context.Bbeans,
Context=prg-all,name=eventRulesEnablement
set the entry messageProcessingExceptionHandler to false.
```

⇒ Corrupt messages will no longer be saved.



#### CAUTION

These settings are not saved permanently in the database. You must re-enter them each time you start the application.

### 3.3.1.3 Change directory for correctly imported HL7 messages

When the system receives an HL7 message that is syntactically correct it saves it to a directory specified for that purpose. The messages are held there for analysis at a later time.

1. Select the **MBeans** tab.
2. Select the MBean with the following name:

```
com.icw.epr:type=Context.MessageReceiver.FileProtocol,
Context=prg-all,name=incomingMessageFileWriter
```

You can make additional settings as indicated in the table in the [Configure storage of corrupt HL7 messages \[page 13\]](#) section.

### 3.3.1.4 Deactivating storage of correctly imported HL7 messages

1. In the MBean, set

```
com.icw.epr:type=Context.Bbeans,
Context=prg-all,name=eventRulesEnablement
set the entry incomingMessageEventFileWriter to false.
```

⇒ Storage of correctly imported HL7-messages is deactivated.



#### CAUTION

These settings are not saved permanently in the database. You must re-enter them each time you start the application.

You can make additional settings as indicated in the table in the [Configure storage of corrupt HL7 messages \[page 13\]](#) section.

### 3.3.1.5 Changing directory for sent HL7 messages

When the system transmits an HL7 message it also saves it to a directory specified for that purpose. The messages are held there for analysis at a later time.



#### NOTE

This function can be used to make messages available to a communications server via a file system interface or FTP interface. The communications server is usually configured for a particular MessageExtension (e.g. \*.dat).

1. Select the **MBeans** tab.
2. Select the MBean with the following name:  
`com.icw.epr:type=Context.MessageSender.FileProtocol,  
Context=prg-all,name=outgoingMessageFileWriter`

For additional settings, please refer to the *Saving corrupt HL7 messages* table in the [Configure storage of corrupt HL7 messages \[page 13\]](#) section.

### 3.3.1.6 Deactivating storage of sent HL7 messages

You can deactivate the storage of sent HL7 messages in the configured directory.

1. In the MBean, set  
`com.icw.epr:type=Context.Beans,  
Context=prg-all,name=eventRulesEnablement`  
set the `outgoingMessageEventFileWriter` to **false**.



#### CAUTION

These settings are not saved permanently in the database. You must re-enter them each time you start the application.

For additional settings, please refer to the *Mbean for saving corrupt HL7 messages* table in the [Configure storage of corrupt HL7 messages \[page 13\]](#) section.

### 3.3.2 Validation settings for data entries

The *Property Configuration* database table contains two properties that define according to the White List which characters are permitted for use in data entries. Find these property setting under the following identifiers:

- `common.validation.legalLdapValueFormatRegex`
- `common.validation.legalStandardInputValueRegex`

The White List for the first-named regular expression is more narrowly limited than the second one. It contains, for example, no umlauts.

The White List of the second regular expression however allows for umlauts. If more characters are required, modify this regular expression:

```
common.validation.legalStandardInputValueRegex
```

Both Regular Expressions have the purpose of preventing external vandalism

### 3.3.3 Setting the number of search results for a patient search

The system will deliver a maximum number of datasets. This enables near real-time responses to patient searches. This number can be set to any desired number.

1. Select the **MBeans** tab.
2. Select the MBean with the following name:  

```
com.icw.epr:type=Context.PersonSearch,  
Context=prg-all,name=personSearchProperties
```

⇒ The details view of the component will open, showing the selected MBean. The *Attributes* MBean is activated.
3. In the **MaxResultCount** field, enter the number of data sets you want to see displayed.



#### NOTE

Note: The **MaxResultCount** field should contain a value between 50 and 100. Higher values impair the system performance and endanger the system stability. Higher values may degrade system performance, lower values may place unreasonable limits on the user.

4. Save your changes by pressing the Enter key.

### 3.3.4 Setting the MSB URL for IHE PIX update notifications

In scenarios where links to patients will be sent to registered systems as IHE PIX Update Notification-conformant messages, you must define a destination URL. You

can adapt the delivery settings of messages via the database system settings, see [Configuration of system properties \[page 57\]](#) section.

You can configure the interface for generated PID messages in the database using `mpi.pix.updateNotification.URL`. Enter the destination URL as configured in the IHE PIX/PDQ module of the MSB. You can activate and deactivate the interface using by setting the value of `mpi.pix.updateNotification.enabled` to **true** or **false**, respectively.

### 3.3.5 Changing an automatic mapping in MPI

When a new patient dataset is added to the system or when a modified dataset is reported they can be mapped to an index patients, either manually or automatically.

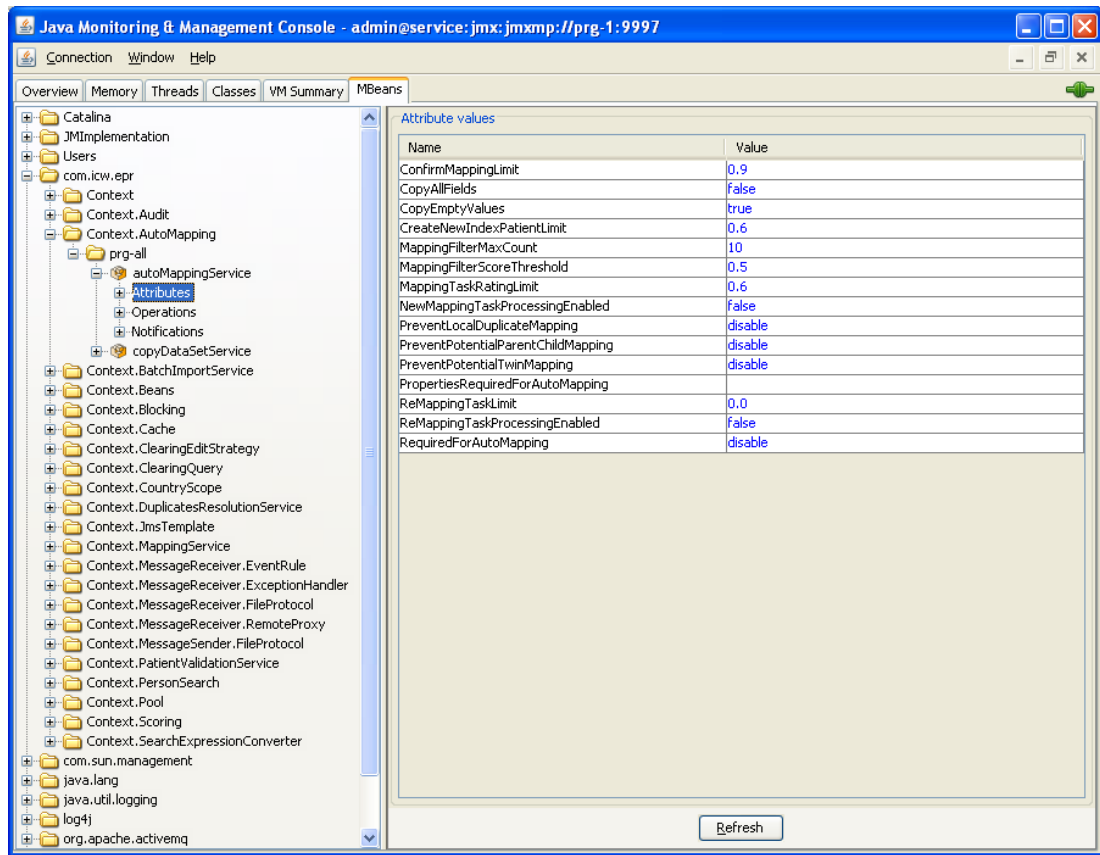
Automatic Mapping is deactivated by default. That means it must be activated explicitly.

Activate mapping of new datasets using the **NewMappingTaskProcessingEnabled** parameter. Thereby the following applies:

- **NewMappingTaskProcessingEnabled:**
  - **True:** Automatic mapping for new datasets is active.
  - **False:** Automatic mapping of new datasets is disabled.

Activate mapping of modified datasets using the **ParameterReMappingTaskProcessingEnabled** parameter. Thereby the following applies:

- **True** to activate automatic processing of modified datasets.
- **False** to deactivate automatic processing of modified datasets.



### 3.3.5.1 Activating automatic mapping and changing parameters

1. Select the **MBeans** tab.
2. Select the MBean with the following name:  
`com.icw.epr:type=Context.AutoMapping,  
Context=prg-all,name=autoMappingService`  
 ⇒ The details view of the component will open, showing the selected MBean. The *Attributes* tab is activated.
3. Select the **NewMappingTaskProcessingEnabled** parameter:
  - **True:** Automatic mapping for new datasets is active.
  - **False:** Automatic mapping of new datasets is disabled.
4. If necessary change other mapping parameters, see [Mapping rules for scoring new datasets \[page 20\]](#) section.
5. Press the Enter key after each change.

#### See also

- [Configure mapping of modified datasets \[page 22\]](#)



### 3.3.5.2 Mapping rules for scoring new datasets

When automatically mapping new datasets to potential index patients the system compares the properties of both datasets according to a defined set of rules. The result of this comparison are mapping decisions with different priorities. The datasets are then mapped according to the highest priority.

Possible decisions made by the system in ascending priority:

- Map new dataset to index patient with the highest degree of similarity (lowest priority)
- Map new dataset to a newly created index patient (medium priority)
- No automatic mapping of the dataset, leave for manual processing (highest priority)

Order of the analysis does not affect scoring.

#### Parameters for configuring rules

There are several rules that can individually be configured using specific parameters:

- **MappingTaskRatingLimit:** Decision based on the complexity of a mapping task. Depending on the number and similarity of the proposed index patient matches the mapping job will be scored as having a greater or lesser potential for error. The MappingTaskRatingLimit [0.0 to 1.0] parameter defines when this rule will trigger manual mapping. If the complexity is lower than the predefined value, the rule will trigger manual mapping.
- **ConfirmMappingLimit** and **CreateNewIndexPatientLimit:** Decision based on the scoring of the most similar index patient. These parameters define the following rules:
  - If the scoring is lower than **CreateNewIndexPatientLimit** a new index patient will be created and the new dataset will be mapped to it.
  - If scoring exceeds **CreateNewIndexPatientLimit** but is less than **ConfirmMappingLimit** the new dataset will be held for manual processing.
  - If scoring exceeds **ConfirmMappingLimit** the dataset will be mapped to an index patient.

Additional rules **PreventLocalDuplicateMapping**, **RequiredForAutoMapping**, **PreventPotentialTwinMapping** and **PreventPotentialParentChildMapping**: Parameters can be applied to set the following states:



- **disabled:** the rule will not be used and therefore does not influence decisions
- **createManualReviewTask:** The new dataset will be processed manually based on the conditions set by the rule.
- **createManualReviewTask:** The new dataset will be mapped to a new index patient based on the conditions set by the rule.
- **PreventLocalDuplicateMapping:** Decision based on source system. If a dataset was already mapped to an index patient with the highest similarity coming from the same source system as the new dataset, this rule will make a decision based on the settings of the PreventLocalDuplicateMapping parameter (state **disabled** , **createManualReviewTask** or **createNewIndexPatient**)
- **RequiredForAutoMapping:** Decision based on the similarity of specific attributes. If certain information of a new dataset does not exactly match the information in the index patient with the highest similarity, the rule will make a decision based on the settings of the **RequiredForAutoMapping** parameter, (state: **disabled**, **createManualReviewTask** or **createNewIndexPatient**). The attributes are defined in the **PropertiesRequiredForAutoMapping** parameter and separated by commas. A single attribute refers to one scoring parameter, see table of scoring parameters in the [Changing Scoring Parameters \[page 35\]](#) section The parameter can for example be set to **<<gender,globalIdentifier1>>**.
- **PreventPotentialTwinMapping:** Decision based on the assumption that a twin was encountered. If the index patient with the highest similarity is already mapped to another dataset that exactly matches the new dataset with regard to last name, date of birth, and address, however the first name does not match, a twin relation is assumed. This rule makes a decision in accordance with the PreventPotentialTwinMapping parameter (state: **disabled**, **createManualReviewTask** or **createNewIndexPatient**).
- **PreventPotentialParentChildMapping:** Decision based on the assumption that a parent-child relationship was encountered. If the index patient with the highest similarity is already mapped to another dataset that exactly matches the new dataset with regard to last name, date of birth, and address, however the first name does not match, a parent-child relation is assumed and this rule decides depending on the settings of the PreventPotentialParentChildMapping parameter (state: **disabled**, **createManualReviewTask** or **createNewIndexPatient**).

### Mapping decision example

A source system receives the data of twins and sends corresponding messages to MPI. The **NewMappingTaskProcessingEnabled** parameter is set to **true**. The **Re-**

**quiredForAutoMapping** and **PreventPotentialParentChildMapping** rules are **disabled**.

Upon receipt of the first message MPI creates a new index patient. When processing the second message, it only candidate it will find is the recently created index patient. Similarities between the datasets is high. The active rules make the following decisions:

- Because there is only one index patient with a high score, the complexity of the new mapping task is very low and the first rule does not affect the decision.
- Because the score is higher than **ConfirmMappingLimit**, the second rule decides that the new dataset will be mapped to the existing index patient.
- The index patient along with the first twin is mapped to a dataset from the same system as the new second twin's dataset. If **PreventLocalDuplicateMapping** is set to **createNewIndexPatient**, the rule will decide to create a new index patient.
- The **PreventPotentialTwinMapping** rule will recognize a potential twin. If **PreventPotentialTwinMapping** is set to **createManualReviewTask**, manual mapping will be triggered.

The decision for manual mapping has the higher priority a manual mapping task will be created.

### 3.3.5.3 Configure mapping of modified datasets

1. In the field **ReMappingTaskProcessingEnabled**, set
  - **True** to activate automatic processing of modified datasets.
  - **False** to deactivate automatic processing of modified datasets.

You can adjust the automatic processing of modified datasets to suit your needs.

To do this, use the following fields:

- **ReMappingTaskLimit**: Assessment score limit for changes to the similarity score between a dataset and the index patients that are mapped to it [ -1.0 to 1.0 ]. If a dataset changes the similarity score will be recalculated.

The old similarity value will be automatically subtracted from the new similarity value. If the entered limit is larger than the result, the mapping will be retained and the task deleted.



#### EXAMPLE

A dataset has degraded by 20% (change: -0.2) and the ReMappingTaskLimit is set to -0.1. Because -0.2 is smaller than -0.1, the mapping will be retained.

---





1. In the field **CopyAllFields**, set:

- **True:** If you want the index patient's data is to be updated each time new and changed datasets are automatically processed.
- **False:** If you want the index patient's data is to remain unmodified when new and changed datasets are automatically processed.

If *CopyAllFields* is active, all data will be transferred to an index patient when new or changed *datasets* are processed. This also includes fields that do not contain any data. This may cause fields that can already contain data to be replaced by empty ones, which may not always be desired. This can be configured globally for all fields:

1. In the field **CopyEmptyValues** set the following options:

- **True:** If you want empty fields also to be copied
- **False:** If you do not want empty fields to be copied

### 3.3.6 Automatic rescoring for insufficient data sets

Most MPI users move deficient, i.e. incomplete data sets to the recycle bin. These data sets can be manually or automatically removed from the recycle bin and restored when they are modified in the source system.

If the automatic rescoring option is activated data sets are removed from the recycle bin when they are modified in the source system. The system matches them to existing index patients and inserts them as new data sets.

1. Select the **MBeans** tab.

2. Select the MBean with the following name:

```
com.icw.epr:type=Context.ClearingEditStrategy,Context=prg-  
all,name=mpiEditService
```

⇒ The details view of the component will open, showing the selected MBean. The **Attributes** tab is activated.

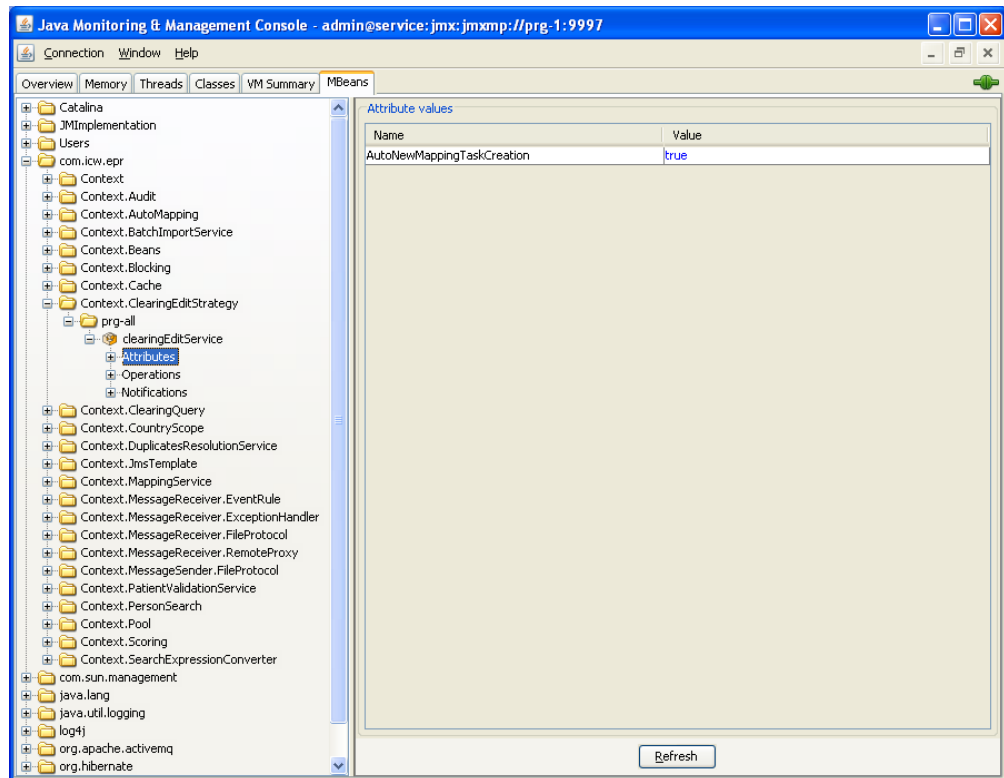


Figure 5: Automatic rescoring for insufficient data sets

3. In the field **AutoNewMappingTaskCreation**, set
  - **True**: when you want data sets in the recycle bin to be rescored automatically.
  - **False**: when you only want data sets in the recycle bin to be restored manually.

### 3.3.7 Setting the sorting order for new matching tasks

In order to keep it within manageable length the *New Matching Tasks* list only contains a limited number of entries. Information about setting the number of entries that will be show can be found in the section [Setting the Number of Patient Search Results \[page 17\]](#). This function puts a limit on access time. This option specifies whether the entries shown will be the oldest or the newest new matching tasks. When users sort this list on their user screen only this limited number of entries will be sorted.

1. Select the **MBeans** tab.
2. Select the MBean with the following name: `Context.ClearingQuery -> prg-all -> clearingQuery`
  - ⇒ The details view of the component will open, showing the selected MBean. The **Attributes** tab is activated.
3. In the **latestFirst** field, select:



- **True:** If you want to retrieve and display the newest new matching tasks from the database
- **False:** If you want to retrieve and display the oldest new matching tasks from the database

The specified sorting order applies also for the list of modified datasets.

### 3.3.8 Suppressing display of data sets

The access right system of MPI allows access rights mapped to roles to be limited to specific organizations. This may mean for example in the context of local editing of mapping tasks that mapping tasks for patient data sets from other organizations will not be displayed in the list.

These organization-related rights can also be expanded to other MPI functions:

- Index patients are only found or created when at least one mapped data set is from the source system of an organization for which the user has the necessary rights. This also applies to the vault and the search when merging index patients.
- Data sets of an index patient from other organizations will be displayed without the source system or organizational rights.
- The hold file list only displays data sets of index patients from permitted organizations

This is how you activate and deactivate these extended access limitations:

1. Select the **MBeans** tab
2. Select the **MBean** with the following name: **com.icw.epr:type=Context.ClearingQuery,Context=prg-all,name=clearingQuery**
  - ⇒ The details view of the component will open, showing the selected MBean. The **Attributes** tab is activated.
3. In the **FacilityPatientDevicesFiltered** field, select:
  - **True:** when the access limitations are to be active as described above
  - **False:** when all information of data sets are to displayed always no matter what the source system is

## 3.4 Advanced setting options

### 3.4.1 Changing the blocking settings

When the MPI looks for similar index patients, it first preselects potential candidates in the database. This preselection process is known as blocking and it accelerates the following scoring stage, see [Scoring settings \[page 34\]](#) section.

The selection algorithms can be defined via JMX. However, please note that the defined blocking algorithm in the database corresponds to an index table containing the pre-calculated blocking values for all index patients. Modifying the blocking algorithms is thus a two-step process.

- Defining algorithms, [see \[page 26\]](#) .
- Recalculating index, [see \[page 31\]](#).



#### CAUTION

Changes to the blocking algorithm should only be made after careful analysis of the prospective patient data. Improper changes can seriously impair the quality of the matching process. They may result in:

- mapping tasks involving an excessive number of candidates
- poor search performance
- incorrectly mapped datasets, because potentially matching index patients failed to be recognized.

#### 3.4.1.1 Specifying the algorithm



#### NOTE

While the algorithm is being modified and the indices are being recalculated, the MPI is in an inconsistent state. Do not shut down the application while recalculation is in progress. Do however suspend the use of the application.

1. Select the **MBeans** tab.
  2. Select the MBean with the following name:  

```
com.icw.epr:type=Context.Blocking,Context=prg-all,  
name=blockingStrategy
```
- ⇒ The details view of the component will open, showing the selected MBean. The *Attributes* tab is activated.

In the field **EncodingAlgorithmDescription** you can define the algorithm as follows:

- Define multiple blocking variables, separated by a semicolon. In the pre-selection process, these variables are linked by logical OR for searching.
- Define attributes using a subset of XPath expressions. Separate subcomponents with a slash (/). Specify search expressions with [`@<attribute>=<value>`]. The usable attributes are described in table 5. Date fields are always returned in the format YYYY-MM-DD.
- Expressions can be linked by a plus sign: (+)
- The following functions can be performed on expressions:
  - `reverse(X)`: read string X in reverse
  - `sort(delimiter, X, Y)`: Sorts alphabetically and combines the character strings X and Y, separated with an underscore character; e.g. `sort(_, "bc", "ab") = "ab_cd"`. This algorithm can be used to reconcile jumbled name elements.
  - `sub(X, start [, length])`: part of string X, beginning at start. If *length* is missing, the entire string from start is returned.
  - `phonet(X, alg)`: Phonetic representation of the character string X. Uses the phonet algorithm (specific to the German language). The parameter alg is either 1 (exact) or 2 (tolerant).
  - `soundex(X)`: Phonetic representation of the character string X. Uses the soundex algorithm. This algorithm is industry standard for phonetic representations of first names, but whose quality is rather questionable.
  - `refinedsoundex(X)`: Phonetic representation of the character string X. Uses the refined soundex algorithm. Equivalent to Soundex, but optimized for spelling corrections.
  - `metaphone(X)`: Phonetic representation of the character string X. Uses the metaphone algorithm. This algorithm is similar to Soundex, but performs better for similarly sounding names.
  - `doublemetaphone(X)`: Phonetic representation of the character string X. Uses the doublemetaphone algorithm. This an improvement over Metaphone.
  - `nysiis(X)`: Phonetic representation of the character string X. Uses the nysiis algorithm. This algorithm is industry standard for phonetic representations of last names.
  - `replace(X, "search expression" [, "replacement expression"])`: In the character string X, replaces all hits for the regular expression *search expression* with the character string *replacement expression*. If *replacement expression* does not exist, all hits for the search expression will be deleted. Both search and replacement expression must be enclosed in double quotation marks.

You can use this function to replace or delete non-alphanumeric characters prior to Blocking or Scoring.

How to handle regular expressions in Java is documented at <http://java.sun.com/javase/6/docs/api/java/util/regex/Pattern.html>.

- `lookup(X, "dictionary" [, "mode"])` : Searches for character string X in the replacement table *dictionary* and replaces it with the corresponding value. The name of the replacement table must be enclosed in double quotation marks. The replacement configuration must be located in the database table *DIC-TIONARY*. The *mode* parameter indicates which parts of the character string will be replaced.
  - *complete*: If the character string consists of exactly one known replacement code in the replacement table, the replacement value will be inserted automatically.
  - *word*: If one word (i.e. a discrete character string enclosed by two empty spaces) is exactly equal to a known replacement code the word will be replaced by the replacement value.
  - *prefix*: If the beginning of a word is exactly equal to one known replacement code in the replacement table the prefix will be replaced by the replacement value.
  - *suffix*: If the end of a word is exactly equal to one known replacement code in the replacement table the end will be replaced by the replacement value.

For more information about the *lookup function*, see [Replacement Tables \[page 31\]](#).

All functions can be nested. Spaces in the expressions are ignored.



#### CAUTION

The total size of the blocking variable may not exceed 4000 characters. If the length of the data fields used in the source system is not limited the length of the partial expressions of the blocking variables should be limited over the `sub()` function. For example, if the first and last name can be longer than 127 characters in the source system you should play it safe by using `sub(name/givenName, 0, 127) + sub(name/familyName, 0, 127)` in place of `name/givenName + name/familyName`.

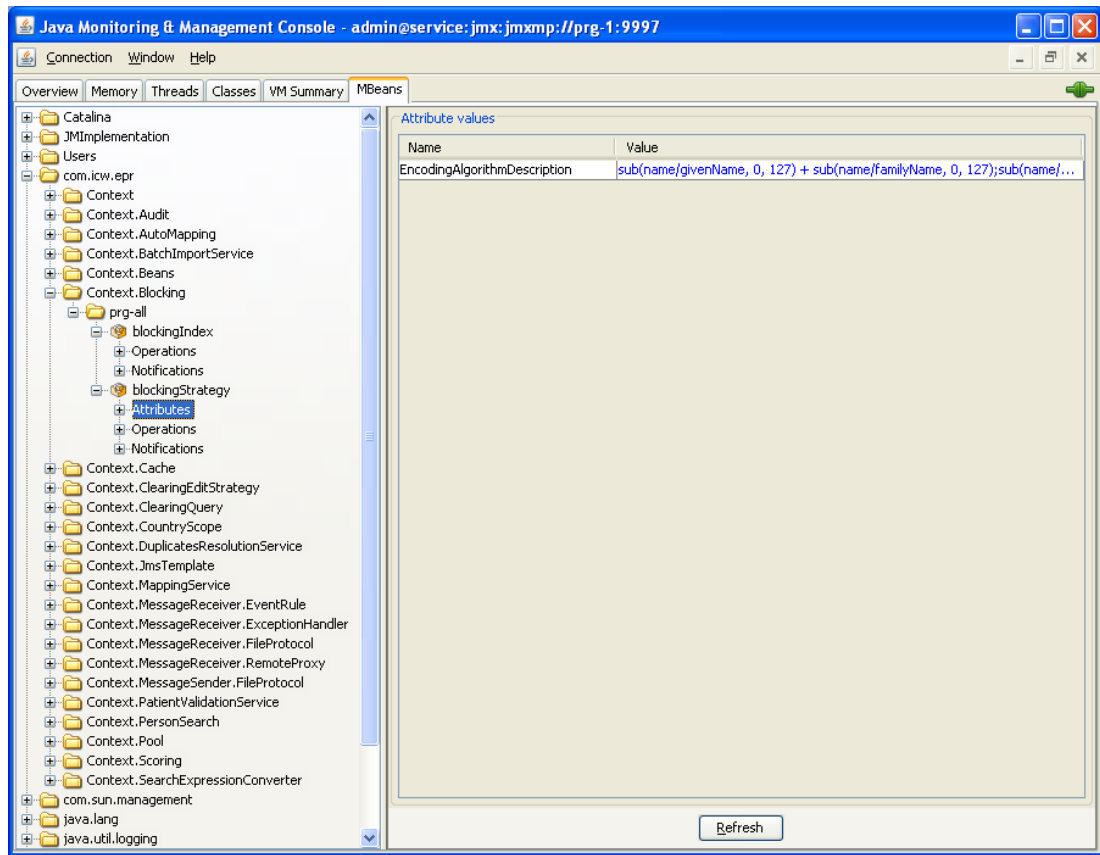


Figure 6: Blocking strategies

### Examples:

Index values for	Expression
„Hans Müller, DOB 31 Jan 1940“, eGK-Nummer=6938658947658456	
hansmüller	name/givenName + name/familyName
hans1940-01-31	name/givenName + birthDate
müller1940-01-31	name/familyName + birthDate
hansnila	name/givenName+ phonet(name/familyName, 2)
hans1940	name/givenName + sub(birthDate, 0, 4)
nila1940	phonet(name/familyName, 2)+ sub(birthDate, 0, 4)
snarel	sub(reverse(name/givenName),0,3) +, sub(reverse(name/familyName),0,3)
6938658947658456	otherIds[@targetType="eGK"]/extension
hans_müller	sort(_, name/familyName, name/givenName)

Index values for „Hans Müller, DOB 31 Jan 1940“, eGK-Numm- er=6938658947658456	Expression
hans19400131	name/givenName + replace (birthDate,"-")
Hxnsmxllxr	replace(name/givenName + na-me/familyName, "[aeiouäöü]", "x")

**Table 4: Sample expressions**

**Attribute values:**

Attribute	XPath expression
First name	name/givenName
Last name	name/familyName
Middle name	name/middleName
Birth date	birthDate (YYYY-MM-DD)
Birth Name	name/birthName
Address	address/streetAddressLine
City	address/city
Postal code	address/zipCode
Country	address/countryCode
Marital status	maritalStatus
Gender	gender
Telephone number (home)	telecom/homePhone
Telephone number (work)	telecom/workPhone
Fax number	telecom/homeFax
e-mail	telecom/homeEmail
Cell phone number	telecom/mobile
Place of Birth	birthPlace/city



Attribute	XPath expression
Social security number (U.S. only)	otherIds[@root='2.16.840.1.113883.4.1']/extension
State (U.S. only)	address/state

**Table 5: Attribute values**

### 3.4.1.2 Recalculating the index



#### CAUTION

Recalculation of the blocking indices will delete and rebuild the indices. During this process – which will normally take less than an hour depending on the size of the database and computer performance – the data of MPI may not be altered through message import or work via the user interface. For this reason, stop the MSB and discontinue work using the user interface.

After the blocking algorithm is modified, the blocking index must be recalculated. This involves recalculating the blocking variables for all index patients.



#### NOTE

Depending on the number of index patients in the database, recalculation may take several hours.

1. Select the **MBeans** tab.
2. Select the MBean with the following name:  
`com.icw.epr:type=Context.Blocking,Context=prg-all,  
name=blockingIndex`
3. Select the **Operations** tab.
4. Click **reset**.
5. To monitor the recalculation's progress click the **status** button.

### 3.4.1.3 Replacement tables

Use the *lookup function* to replace partial character strings with an entry stored in the database table DICTIONARY. This function can be used to support a multitude of use cases where blocking or scoring must be modified or improved to accommodate local conditions. For example, you have options to:

- Replace occasionally used nicknames with the corresponding full first name, such as **Robert** in place of **Bob**
- Replace variant spellings in addresses, especially for street names, with normal spellings, for example High St instead of High Street
- Replace abbreviations with full expressions, for example First instead of 1st

If the *lookup function* is used when blocking, additional candidate index patient whose names or addresses contain spelling variations will be found and included for scoring purposes.

When the *lookup function* is used in scoring, the scores of fields that contain variant spellings will be higher. The results in better differentiation in scores for fields whose contents truly differ.

The *mode parameter* can be used to influence whether and how partial character strings will be replaced. The replacement table DICTIONARY contains the following entries as an example:

ID	dictionary	key	normalized
0	street	str.	straße
1	street	Str	straße
2	street	strt	street
3	street	st	street
4	street	nw	northwest

The following table clarifies the workings of the different replacements modes for the function `lookup(address/streetAddressLine, "street", <mode>)`

<mode>	Industriestr. 12	43 Industry Strt nw
"complete"	industriestr. 12	43 industry strt nw
"word"	industriestr. 12	43 industry <b>street northwest</b>
"prefix"	industriestr. 12	43 industry <b>street northwest</b>
"suffix"	industriestr <b>asse</b> 12	43 industry <b>street northwest</b>

It can be seen that

- Character strings are always transformed into lower case.
- Partial character strings that do not begin or end a word are never replaces (Industriestr**asse**, Indust**ry** Street).
- All word replacements may also be identified as prefixes or suffixes. The stand-alone word **strt** will be replaced in *word* mode, but also in the *prefix* and *suffix* modes.

- When making replacements, always the longest prefixes and suffixes in the table will be used. The character string **industry strt** will not be replaced by **industry street** in *prefix* mode, even though a matching replacement entry for **st** exists.
- The original value, in lower-case characters, will be returned if no matching replacement exists.
- Multiple replacements may occur within one character string if more than one replacement is found.

The replacements tables may be read in at system start. The system must be restarted for changes to the configuration to become active. The *dictionary* given in the *lookup function* corresponds to the column of the same name in this database table.

It is often helpful to combine the *lookup function* with the *replace function*, for example

```
lookup (replace (address/streetAddressLine, "\."), "street", "suffix")
```

first deletes the (abbreviation) point and then replaces the resulting character string with the matching table entry.

Translation tables can be prepared in advance and copied to the database when the MPI solution is installed by inserting them into the file `test-data.xml`:

```
...
<dictionary>
  <entry dictionary="nickname" key="a.b." normalized="abiah" />
  <entry dictionary="nickname" key="ab" normalized="abel" />
  <entry dictionary="nickname" key="abbie" normalized="abigail" />
  <entry dictionary="nickname" key="abby" normalized="abigail" />
  <entry dictionary="nickname" key="al" normalized="allen" />
  ...
  <entry dictionary="street" key="st." normalized="street" />
  <entry dictionary="street" key="st" normalized="street" />
  <entry dictionary="street" key="strt" normalized="street" />
  <entry dictionary="street" key="ave" normalized="avenue" />
  ...
</dictionary>
```



#### NOTE

#### Value for key

In the current version you can give just one single *key* for each normalized value. For example, it is not possible to replace **al** with both **alan** and **allen**. Conversely however, multiple *keys* may be replaced by the same normalized value, for example **st**, **st.**, or **strt** for **street**.

## 3.4.2 Scoring settings

### 3.4.2.1 Changing the color coding for scoring values

When scoring, the MPI calculates a quantitative score for the match between the demographic data of two patients. The score is expressed as a percentage between 0% (no match) and 100% (complete match). Very low scores are unlikely, as the pre-selection process during blocking will usually have loaded similar datasets from the database, see [Changing the blocking settings \[page 26\]](#) section.

Aside from the percentage score, the scores are also indicated qualitatively in the form of colored bars. By adjusting the scoring function, you can control the limits used to display the scores of a new dataset in color.

1. Select the **MBeans** tab.
2. Select the MBean with the following name:

```
com.icw.epr:type=Context.Scoring,Context=prg-all,  
name=scoringStrategy
```

⇒ The details view of the component will open, showing the selected MBean. The **Attributes** tab is activated.

Scores higher than the value of the attribute **HighRatingLimit** are marked in green. All scores with values between the **HighRatingLimit** and **LowRatingLimit** attributes are marked in yellow. Scores from mapping tasks lower than the attribute **LowRatingLimit** are marked in red.

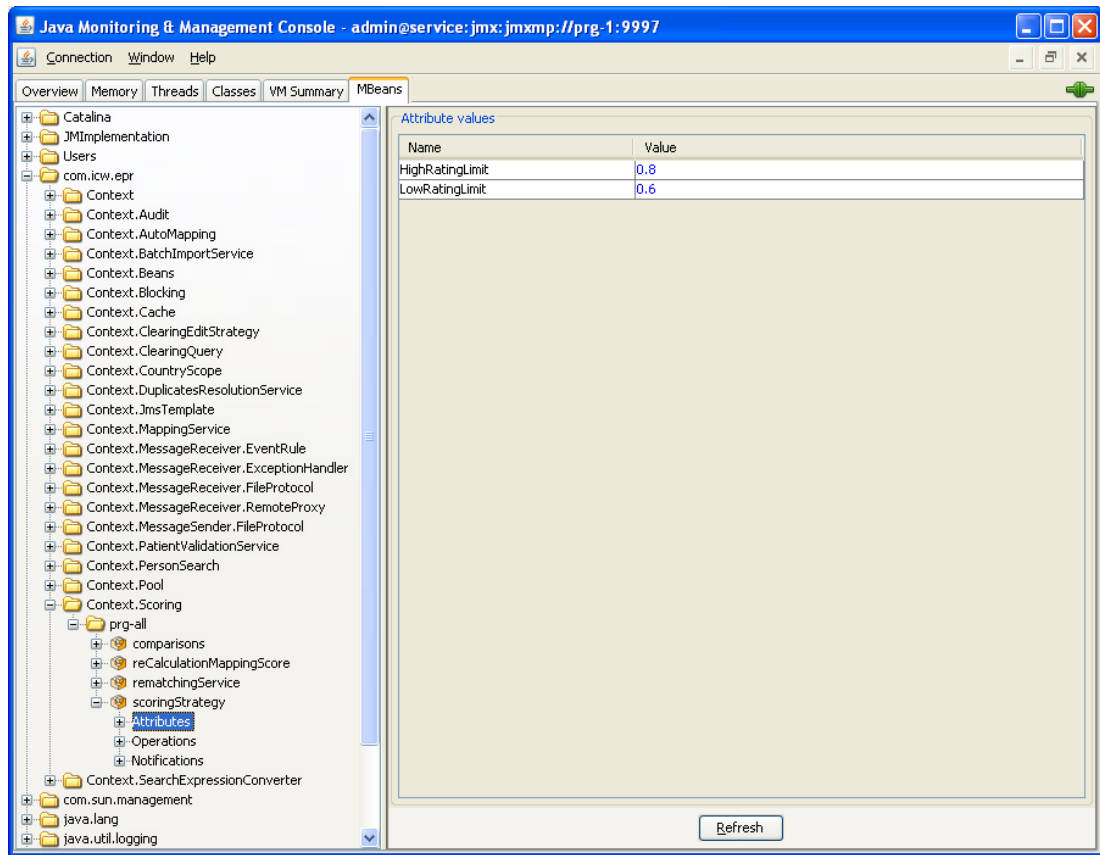


Figure 7: Scoring

### 3.4.2.2 Changing scoring parameters

Each time a patient is registered, a new dataset is generated. These can be processed automatically if required, see [Changing an automatic mapping in the MPI \[page 18\]](#) section.

The database is searched for matching index patients for each new dataset and when a mapping is dissolved. The system will return a list of hits as potential candidates, see [Changing the Blocking settings \[page 26\]](#) section.

When mapping, relevant patient datasets are read, weighted and scored.

The following describes options for adapting the scoring parameters. The result is a similarity score.

The following types of comparisons are used. These are distinguished by the *type* parameter and cannot be changed:

- **exact**: The system checks whether the attributes are identical.
- **fuzzyString**: A Jaro-Winkler algorithm is applied to test attributes for lexical similarity (recommended for names only).

- **fuzzydate:** A similarity algorithm is applied to test attributes for dates. This includes provisions for a deviation of several days, transposition of day and month, and for deviant month (recommended for date entries only)
- **contains:** Checks whether a shorter form of an attribute is contained in a longer value
- **editDistance:** A Levenstein algorithm is applied to test for lexical similarity (recommended for IDs only)
- **frequency:** Attributes are tested for identity, but identity is scored on the basis of a frequency table, see [Changing the filtering \[page 44\]](#) section.

The following setting options are available for all datasets included in the scoring:

- **encodingAlgorithm:** [1] This expression returns a value which is either a direct attribute of, or calculated from, the patient dataset. The expressions used are also used in blocking, see [Specifying an Algorithm \[page 26\]](#) section.
- **enabled:** [2] Defines whether this comparison is included in the score calculation.
- **optional:** [3] Defines whether this comparison is optional in the score calculation. In that case, only a non-identical attribute will affect the scoring. If the attribute is missing in the dataset or for the index patient, it will be counted as a match.



#### NOTE

The **optional** parameter should not be used with MPI Version 2.8 or higher, because **missingValueStrategy** offers better setting options. It is still supported, however.

- **missingWeight:** [4] This parameter is only used with MPI Version 2.8 when **missingValueStrategy** is set to **none** or **weight**. It represents the weighting factor when an attribute is missing in two datasets that are being compared. We recommend leaving this at 0.0, since in most cases attributes are missing because data is lacking. In this case, the mere fact that attributes are missing does not in itself constitute information.
- **u:** [5] Probability that this attribute is identical in two compared datasets when they do not represent the same person [ 0 .. 1 ]. The lower this value, the stronger the effect on the overall scoring if the two attributes are identical/not identical. Frequently, this value reflects the limited number of possible values, for instance when they are narrowly distributed, such as `birthMonth=1/12=0.8333`.
- **m:** [6] Probability that this attribute is identical in two compared datasets when they represent the same person [ 0 .. 1 ]. The higher this value, the stronger the effect on the overall scoring if the two attributes are identical/not identical. This value can



often only be estimated. Usually an error or deviation probability will be subtracted from the value 1, for example `city=1-20%=0.80` would predict that one in five patients have moved. The value `familyName` often changes due to marriage. This value should be lower than the value for `givenName`, which usually does not change, but which may have been recorded incorrectly.

- **minimumSimilarity:** [7] : Lower limit for the results of the similarity algorithm, below which two datasets should be considered to be different. This setting option is only available for `fuzzyString` comparisons. The value entered must be between 0 and 1. If the value calculated by the fuzzy string comparison is greater than the value entered here, the corresponding will be considered similar, if not it will not be considered similar.
- **missingValues:** [8] A comma-separated list of values that are considered missing. The default setting is for character strings with a length of 0 will be considered missing and so do not need to be given separately. This setting can also be used to exclude dummy or test values from scoring.
- **missingValueStrategy:** [9] Defines behavior when values are missing. Following values are allowed:
  - **agree:** Scores the missing value as identical
  - **disagree:** Scores the missing field values as non-identical
  - **average:** Scores missing field values neutrally, i.e. as an average between identical and non-identical
  - **compare:** Performs comparison despite missing values
  - **skip:** Ignores the comparison of the fields, i.e. behaves as though **enabled** were set to false
  - **weight:** Applies the weighting factor from **missingWeight**
  - **none:** Applies the weighting factor from **missingWeight**, but also applies the setting **optional**. Using **none** achieves the same behavior as MPI Versions 2.7.1 and lower, so this is the default setting.
- **frequencyTable:** [10] URL of a frequency table. This setting option exists only for `frequency` comparisons. See [Changing the filtering \[page 44\]](#) section.
- **maxFrequencyAgreementWeight:** [11] Maximum weight for comparisons in a frequency table. This setting option exists only for **frequency** comparisons. See [Changing the filtering \[page 44\]](#) section.
- **daysTolerance:** [12] Maximum deviation from an exact date for which a comparison scoring will be performed, in days. This setting option is only available for `fuzzyDate` comparisons.



### CAUTION

Changes to these values should be made only after careful analysis of the expected patient data. Improper changes can seriously impair the quality of the matching process.

This could lead, for example, to errors in manual or automatic mapping of patients.

### List of scoring parameters

The numbers [1-9] in the heading of the table refer the settings options for datasets mentioned above: **encodingAlgorithm**: [1], **enabled**: (TRUE/FALSE) [2,] **optional**: (TRUE/FALSE) [3], **missingWeight**: [4], **u**: [5], **m**: [6], **minimumSimilarity** [7], **missingValues** [8], **missingValueStrategy** [9].

Parameter	1	2	3	4	5	6	7	8	9
birthDay	sub(birthDay, 8,2)	T	F	0.0	0.002731	0.95			none
birthMonth	sub(birthDate, 5,2)	T	F	0.0	0.08333	0.95			none
birthYear	sub(birthDate, 0,4)	T	F	0.0	0.01250	0.95			none
birthDate	birthDate	F	F	0.0	0.00003	0.95			none
birthName	birthName	T	F	0.0	0.01	0.99	0.7		none
birthPlace	birthPlace/city	F	F	0.0	00.005	0.99	0.85		none
city	address/city	T	F	0.0	00.005	0.80	0.85		none
country	address/ countryCode	T	F	0.0	0.30	00.995			none
familyName	name/ familyName	T	F	0.0	0.0010	0.85	0.7		none
gender	gender	T	F	0.0	0.5	0.99		U	none
givenName	name/ givenName	T	F	0.0	0.01	0.95	0.7		none
maritalStatus	maritalStatus	F	F	0.0	0.5	0.85			none
middleName	name/ middleName	F	T	0.0	0.01	0.95	0.7		none



Parameter	1	2	3	4	5	6	7	8	9
homePhone	telecom/ homePhone	F	F	0.0	0.0020	0.85			none
homeFax	telecom/ homeFax	F	T	0.0	0.0020	0.85			none
homeEmail	telecom/ homeEmail	F	T	0.0	0.0020	0.85			none
workPhone	telecom/ workPhone	F	T	0.0	0.0020	0.85			none
mobile	telecom/mobile	F	F	0.0	0.0020	0.85			none
streetAddress	address/ streetAddress Line	T	F	0.0	0.0050	0.8	0.85		none
zipCode	address/ zipCode	T	F	0.0	0.0010	0.85			none
globalIdentifier1	otherIds[ @root = '2.16 .840.1.113883 .3.37.4.1.5.2 ' ]/extension	F	F	0.0	00.999	0.000001			none
globalIdentifier2	otherIds[ @root = '2.16 .840.1.113883 .3.37.4.1.5.3 ' ]/extension	F	F	0.0	00.999	0.000001			none
globalIdentifier3	otherIds[ @root = 'rootValue' ]/extension	F	F	0.0	00.999	0.000001			none
globalIdentifier4	otherIds[ @root = 'rootValue' ]/extension	F	F	0.0	00.999	0.000001			none

Parameter	1	2	3	4	5	6	7	8	9
globalIdentifier5	otherIds[ @root = 'rootValue' ]/extension	F	F	0.0	00.999	0.000001			none
state	state	F	F	0.0	0.1	0.99			none

**Table 6: Scoring Parameters**

In addition to these scoring parameters, there are additional sets of parameters that can include source system-specific parameters, see [Source system specific configuration \[page 41\]](#) section. [\[page 41\]](#)



**NOTE**

The root entered for Global Identifier 1 defines the name space for the patient for the system configured in MPI administration. These settings may differ depending on the solution (for example, the root might have an AHV number), and accordingly may have different values for u and m.

To display and map such identifiers a corresponding system, for instance for social security number or AHV number, must be created within the application. (See the Administration Manual).

### 3.4.2.3 Frequency tables

Frequency tables can be used when values for an attribute are expected to be very unequally distributed so that their weight when they are identical or non-identical varies. It be used for example to give more weight to two identical family names when they are rare than when they are common. This applies to places of residence, where many more patients can be found in major cities near the medical facility so the effect on the overall score should be less than for patients who live in small villages in the surrounding countryside.

Depending on the number of other comparisons and/or their configuration the use of frequency tables can reduce the effect on the overall score when the most common values recur, but seldom more than by 5 percent.

To subject an attribute to frequency-independent scoring perform the following steps:



1. Set the comparison **type** to `frequency`
2. In a file `/<absolute-path/to/frequency/file.properties>` create a frequency file in Java Properties format, i.e. one entry per line `<value>=<frequency>`. Lines that begin with a hash mark (#) will be ignored and can be used for remarks. All values **must** be written in lower-case characters. If the table only contains the most common values a special value `__sum__` should be inserted that gives the dimension of reference basic total, otherwise the frequencies will simply added to each other. If relative frequencies  $0 < h < 1$  are included in the table `__sum__` should have the value 1. Names that do not occur in the table will be scored as exact comparisons without the table.
3. In **frequencyTable**, insert a reference to the frequency table. The format is:`</absolute-path/to/frequency/file.properties>`.
4. In the field **maxFrequencyAgreementWeight** enter the maximum weight that should be applied to this comparison. The standard value 0.0 will be replaced by the weight when the degree of similarity is "normal".

#### Examples:

You want to use a frequency table for last names. This is done by setting the comparison for `name/familyName` to `frequency`, letting `m` equal 0.85 and `u` equal 0.001. The frequency table holds 9000 different names, the most common of which having a frequency of 2400, the least common one a frequency of 1. The sum of the frequencies amounts to 79731 names in all

When a frequency table is not used an absolute weight of 9.73 is applied, and negative weight of -2.73 for non-matches. With frequency table, a match for the most common names give a weight of only 5.05, the rarest names however a weight of 16.24. If **maxFrequencyAgreementWeight** remains set to 0.0, the weight for rare names is reduced to 9.73. When a frequency table is used a negative weight of -2.73 is awarded for non-matches.

Distribution may vary even more widely for places of residence (`address/city`). For example, if 50 percent of all patients live in the same city a match would only have a weight of 1.0 compared to the standard setting in MPI, which is 7.3.

#### 3.4.2.4 Source system-specific configuration

Quality of demographic data managed by different source systems can vary. However, the standard parameters will use the same methods to compare and score data for all systems. You can define source system-specific settings to allow for specific or non-standard parameters other than those described in the [Changing scoring parame-](#)

ters [page 35] section for individual attributes of a dataset. To do so, the following two settings are available for all parameters:

- **replaces**: references the parameter that is to be replaced for datasets of a certain source system. Parameter names from the Scoring parameters table are valid, for example *gender*.
- **dataSourceId**: Contains the Patient-OID of the source system. Specific settings are defined for data from this source system.

Parameters containing these settings are source system-specific parameters. A separate set of unused parameters(sourceSpec01 to sourceSpec10) prevents standard parameters from being modified. The remaining settings match the standard parameters and define scoring of the comparison.

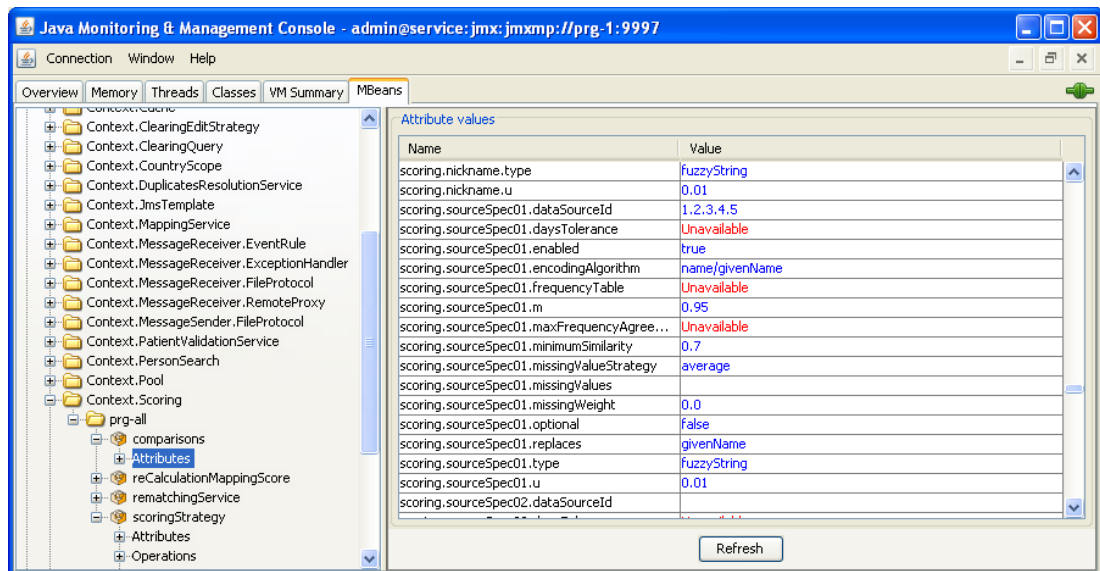


Figure 8: Source system-specific configuration

## Example

In the example the quality of the leveraged first names is very high. Therefore, the standard parameter assesses the data for exact similarity. It is defined as follows:

Setting	Value
scoring.givenName.encodingAlgorithm	name/givenName
scoring.givenName.type	exact
scoring.givenName.enabled	true
scoring.givenName.optional	false
scoring.givenName.m	0.95

Setting	Value
scoring.givenName.u	0.01
scoring.givenName.missingWeight	0.0

**Table 7: Standard parameter for scoring.givenName**

Data of source system A with the patient OID 1.2.3.4.5 contains many typing errors in first names. When comparing similarity between datasets from this system and index patients, it would be advisable to use a fuzzy search as opposed to searching for exact matches, i.e. **fuzzystring** should be used for type instead of **exact**. The following describes the additional settings you have to configure:

Setting	Value
scoring.sourceSpec01.encodingAlgorithm	name/givenName
scoring.sourceSpec01.type	fuzzyString
scoring.sourceSpec01.enabled	true
scoring.sourceSpec01.optional	false
scoring.sourceSpec01.m	0.95
scoring.sourceSpec01.u	0.01
scoring.sourceSpec01.missingWeight	0.0
scoring.sourceSpec01.minimumSimilarity	0.7
scoring.sourceSpec01.replaces	givenName
scoring.sourceSpec01.dataSourceId	1.2.3.4.5

**Table 8: Replacing the parameters in scoring.givenName**

The source system-specific parameter replaces the standard parameter. No settings are carried over. This also applies to the **enabled** setting. In order for the specific parameter to be active, **enabled** must be set to **true**.



**NOTE**

**Using specific parameters to deactivate standard parameters**

Setting **enabled** of a specific parameter to **false** will disable a standard parameter.

A specific parameter cannot replace another specific parameter. Defined **replaces** and **dataSourceId** settings will tell the system that the parameter is a specific parameter.

### 3.4.2.5 Recalculating scoring values

Adjusting the scoring parameters will only affect recalculated scoring values. Already calculated scoring values will not be changed by this initially. Manual recalculation will allow you to calculate all scores with the new parameters.

1. Select the **MBeans** tab.
2. Select the MBean with the following name:  
**com.icw.epr:type=Context.Scoring,Context=prg-all,name=reCalculation-MappingScore**  
⇒ The details view of the component will open, showing the selected MBean. The Operations tab is activated.
3. To restart calculation click the **startRescore** button.
4. To monitor the recalculation's progress click the **recalculationStatus** button.
5. To pause the recalculation click the pause button.
6. To continue the recalculation process click the **resumeRescore** button.

### 3.4.2.6 Repeat automatic mapping

Oftentimes a review of the remapping tasks after recalculating the scores or changing the configuration of automatic mapping (see section 3.3.5) will lead to a number of these being able to be processed automatically thereby minimizing manual efforts.

1. Select the **MBeans** tab.
2. Select the MBean with the following name:  
**com.icw.epr:type=Context.Scoring,Context=prg-all,name=rematchingService**
3. To restart calculation click the **startRematching** button
4. To monitor the recalculation's progress click the **rematchingStatus** button
5. To pause the recalculation click the **pause** button
6. To continue the recalculation process click the **resumeRematching** button

## 3.4.3 Changing the filtering



#### CAUTION

Changes to these filter settings should be made only after careful analysis of the expected patient data. Improper changes can seriously impair the quality of the matching process.



Once the two previous steps of blocking and scoring are complete the mappings suggested in the new datasets are filtered. Low scores are eliminated to limit the number of suggestions.

1. Select the **MBeans** tab.

2. Select the MBean with the following name:

```
com.icw.epr:type=Context.AutoMapping,Context=prg-all,name=auto-  
MappingService
```

3. Define the filter threshold in the **ScoreThreshold** attribute.

Candidates with scores lower than this threshold will be filtered out and not displayed.

The **MaxCount** attribute additionally defines the maximum number of suggestions per new data set.



#### NOTE

If more candidates than **MaxCount** are proposed whose scores are higher than ScoreThreshold, those with the lowest scores will be filtered out.

Name	Value
ConfirmMappingLimit	0.9
CopyAllFields	false
CopyEmptyValues	true
CreateNewIndexPatientLimit	0.6
MappingFilterMaxCount	10
MappingFilterScoreThreshold	0.5
MappingTaskRatingLimit	0.6
NewMappingTaskProcessingEnabled	false
PreventLocalDuplicateMapping	disable
PreventPotentialParentChildMapping	disable
PreventPotentialTwinMapping	disable
PropertiesRequiredForAutoMapping	
ReMappingTaskLimit	0.0
ReMappingTaskProcessingEnabled	false
RequiredForAutoMapping	disable

### 3.4.4 Copying missing values

When automatically processing new or changed datasets the option exists to suppress the copying of individual fields that contain standardized or nominal content, for example when they are defined as required fields by the source system but the corresponding information was not available at the time. Examples might include date of birth (Jan 1, 1800 for example), or social security number (999-99-9999). Field content of this kind can be interpreted as "missing value" to prevent their being copied to the index patient's record. This is done in the following manner:

1. Select the **MBeans** tab
2. Select the MBean with the following name:  
`com.icw.epr:type=Context.Automapping, Context=prg-all,name=copy-DataSetService`  
⇒ The details view of the component will open, showing the selected MBean. The **Attributes** tab is activated.

It is possible to define values for a total of six fields you want to have interpreted as missing.

The affected field is defined in the field `autoMapper.missingValue.xpath.expression.<n>`. The notation corresponds to the `encodingAlgorithm` entry for scoring, see [Setting Scoring Parameters \[page 35\]](#).

The value you want to be interpreted as missing is entered into the field `autoMapper.missingValue.xpath.value.<n>`. If you want to enter multiple values, separate them with a comma.



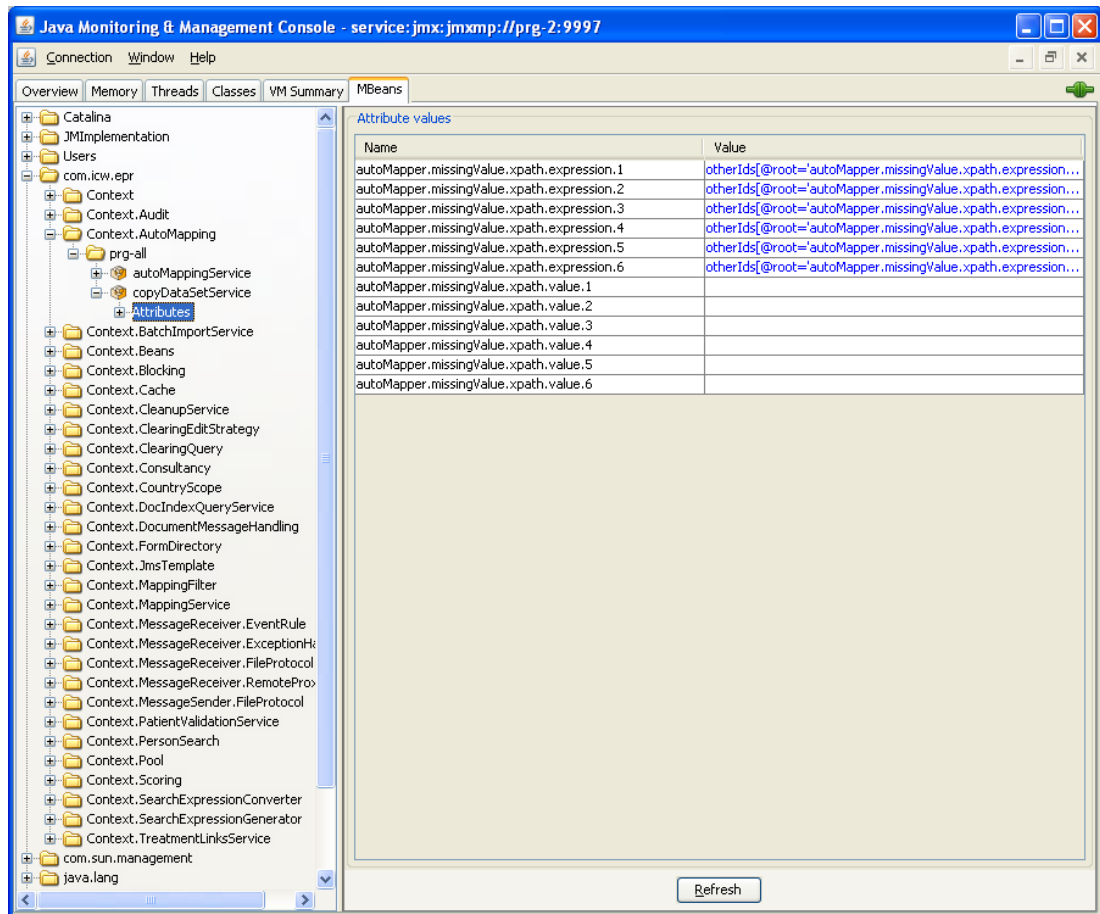


Figure 9: MBean, settings to define missing values

- Example 1:** Do not copy certain dates (Jan 1, 1800 and Jan 1, 1900):
 

```
autoMapper.missingValue.xpath.expression.1=birthDate
autoMapper.missingValue.xpath.value.1=1800-01-01,1900-01-01
```
- Example 2:** Do not copy social security number 999-99-9999
 

```
autoMapper.missingValue.xpath.expression.2=
otherIds[@root='2.16.840.1.113883.4.1']/extension
autoMapper.missingValue.xpath.value.2=999-99-9999
```

In the scoring configuration, *missingValues* awards special weights to missing values or to values that are *interpreted* as missing, see [Setting Scoring Parameters \[page 35\]](#). These Values consists of partial field entries or of a combination of multiple fields. That means that this configuration is not appropriate to permit or forbid the copying the entire contents of fields during automatic processing for new or changed datasets. That is the reason why the settings of *autoMapper.missingValue* are used independently of scoring to control the copying of field of fields to an index patient.



### 3.4.5 Changing the search expression converter

This section defines the settings used with searches.

1. Select the **MBeans** tab.
2. Select the **MBean** with the following name:  

```
com.icw.epr:type=Context.SearchExpressionConverter,Context=prg-all,name=<Attribute name from table 7>
```
3. In the *wildcardsAccepted* attribute, specify whether wildcards are permitted for use as search parameters:

Attribute name	Standard value	Description
birthNameSearchExpressionConverter	TRUE	Defines whether search terms with placeholders are permitted for birth name.
caseIDSearchExpressionConverter	FALSE	Defines whether search terms with placeholders are permitted for the administrative case number.
citySearchExpressionConverter	TRUE	Defines whether search terms with placeholders are permitted for the town/city.
countryCodeSearchExpressionConverter	FALSE	Defines whether search terms with placeholders are permitted for the country code.
familyNameSearchExpressionConverter	TRUE	Defines whether search terms with placeholders are permitted for last name.
givenNameSearchExpressionConverter	TRUE	Defines whether search terms with placeholders are permitted for first name.
patientIdExtensionSearch ExpressionConverter	FALSE	Defines whether search terms with placeholders are permitted for the patient number.

Attribute name	Standard value	Description
patientIdRootSearchExpressionConverter	FALSE	Defines whether search terms with placeholders are permitted for a range of patient numbers.
streetAddressLineSearch ExpressionConverter	TRUE	Defines whether search terms with placeholders are permitted for the street address.
zipCodeSearchExpressionConverter	TRUE	Defines whether search terms with placeholders are permitted for the zip code.
indexPatientIdExtensionSearch Expression-Converter	FALSE	Defines whether search terms with placeholders are permitted for the index patient ID.
indexPatientIdRootSearch ExpressionConverter	FALSE	Defines whether search terms with placeholders are permitted for index patient numbers.
middleNameSearchExpressionConverter	TRUE	Defines whether search terms with placeholders are permitted for a middle name.

**Table 9: Attributes of the Search Expression Converter**

### 3.4.6 Automatically merge index patients

It is possible that a faulty legacy data import and a faulty blocking strategy result in new index patients being created automatically although mapping to an existing index patient would be correct. These redundant index patients match in all or almost all attributes. Manually merging these index patients however is oftentimes too time-consuming because of the large number of duplicates. In these cases, use the function as follows:

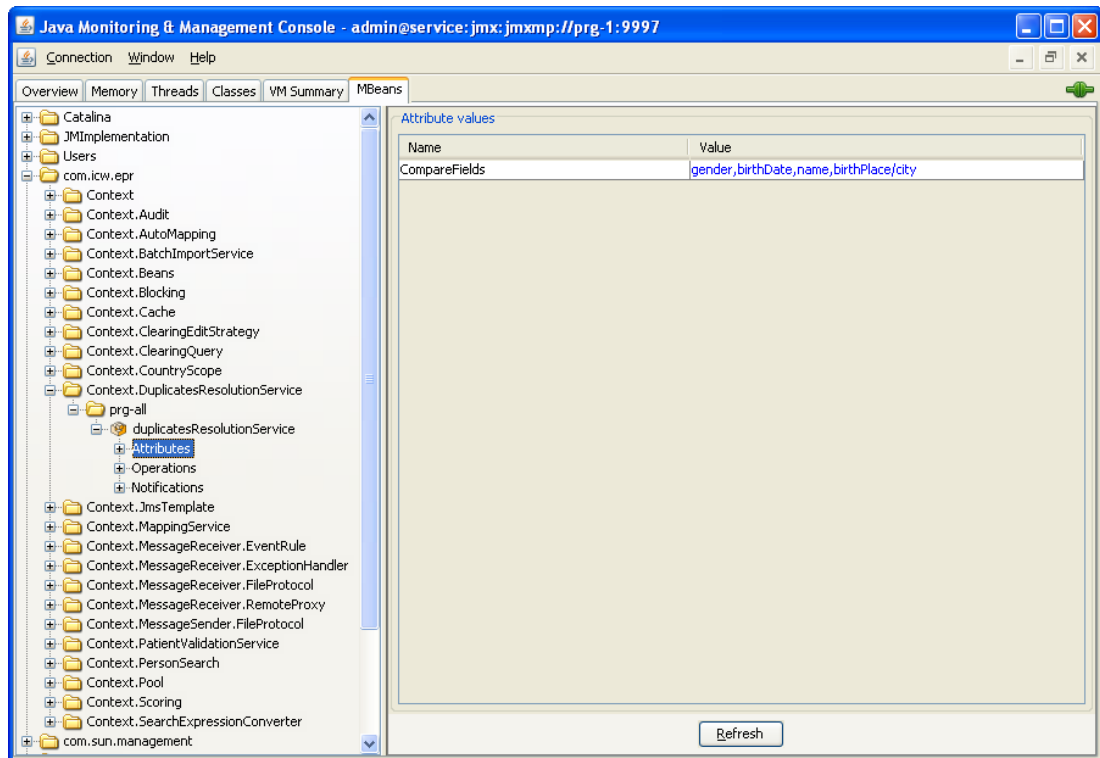


Figure 10: Automatically merge index patients

1. Select the **MBeans** tab
2. Select the MBean with the following name:  
**com.icw.epr:type=Context.DuplicatesResolutionService,Context=prg-all,name=duplicatesResolutionService**
  - ⇒ The details view of the component will open, showing the selected MBean. The **Attributes** tab is activated.
3. In the **Compare Fields** field, enter which fields of the two index patients have to be identical to be considered duplicates and are to be merged. The predefined setting is **gender,birthDate,name,birthPlace/city**, indicating that gender, date of birth, all parts of the name and the place of birth have to be identical.
4. Select the **Operations** tab
5. To start the process click **Start**
6. To monitor the progress click the **status** button

The MPI will send a merge message to the MSB for each pair of index patients merged. This allows PIX update subscribers watch mapping of index patients and data sets.

### 3.4.7 EHCache Configuration

The cache that will be used when documents are called from source systems can be configured to the system's scale or stage of growth.



Modify the configuration directly in the file ehcache.xml under

```
<<CATALINA_HOME>>/prg-all/webapp/<<webapp-name>>/WEB-INF/classes.
```

Any needed changes must be made to the cache Element named `documentCache`.

- **maxElementInMemory** (default: 0) defines the maximum number of elements that will be generated in the memory.
- **Eternal** (default: false) defines whether or not an element will reside permanently in memory.
- **overflowToDisk** (default: true) defines whether elements will be written to the hard disk drive when the internal memory cache reaches the value `maxInMemory`.
- **timeToIdleSeconds** (default: 8 days) defines how long element will remain in the cache when called unless not called again.
- **timeToLiveSeconds** (default: 14 days) defines how long an element will remain in the cache before it becomes invalid.
- **diskPersistent** (default: true) defines whether elements will persist on the hard disk drive when the virtual machine is restarted.
- **diskExpiryThreadIntervalSeconds** (2 days) defines the time period after which the disk expiry thread will delete invalid documents.
- **memoryStoreEvictionPolicy** (LRU) defines the strategy for deleting documents from memory.

The settings are carried over at the next restart. The contents of the cache are backed up when shutting down and read in again at startup.

## 3.5 Logging control

The most important logging components in the system are those of the types `Logger` and `Appender`.

- **Logger:** A logger categorizes and filters log events. It sends them to the corresponding appender.

Each logger has a unique name that refers to the associated system component. The name uses the naming scheme for fully qualified identifiers. This corresponds to the class name with a package name affix separated by a period. Loggers thus form a hierarchical name space.

- **Appenders** use a defined layout to convert a log entry to a matching text. This text is then written in a log file.

Use the appender to determine the log file and to define the rotation strategy for the logged events. Usually, both the MBean for the appender itself, as well as the MBean for the layout used by the appender, are assigned to the appender.

Loggers, appenders, and layouts are each represented in their own MBeans. The names are formed according to the following naming scheme:

- MBeans for loggers:

```
<<webapp-name>>-logging:logger=<logger name>
```

<logger name> is the name of the logger represented by the MBean.

- MBeans for appenders:

```
log4j:appender=<appender_name>
```

<appender\_name> is the name of the appender represented by the MBean.

1. Select the **MBeans** tab.
2. In the form, select the MBean that corresponds to the Logger, Appender or Layout.
3. Enter your changes.

For a list of attributes belonging to loggers, appenders, and layouts, see [Tables \[page 57\]](#) section.

### 3.5.1 Setting the Log Level

1. Select the **MBeans** tab.
2. In the form, select the MBean that corresponds to the *logger*.  

```
<<webapp-name>>-logging:logger=<logger name>
```

⇒ The details view of the component will open, showing the selected MBean. The *Attributes* tab is activated.
3. In the **priority** field, enter the desired log level. Select a constant from the set {DEBUG, INFO, WARN, ERROR, FATAL}.



#### NOTE

Use only these constant given here.

4. Press the **Enter key** after making changes.

### 3.5.2 Setting the Appender

1. Select the **MBeans** tab.
2. In the form, select the MBean that corresponds to the appender:  

```
log4j:appender=<appender name>
```

⇒ The details view of the component will open, showing the selected MBean. The *Attributes* tab is activated.



3. You can modify the following attributes. For some appenders the range of editable attributes is limited.

- **threshold:** Specifies the degree of detail of the log trace. Select a constant from the set {DEBUG, INFO, WARN, ERROR, FATAL}, or do not enter anything at all.
- **file:** File system path for the log file.
- **datePattern:** Pattern for the log file date rotation.
- **encoding:** Character encoding of the log file.
- **append:** Defines the behavior of the appender when the system is started:
  - **True:** The appender adds log events to an existing log file.
  - **False:** The appender deletes existing log files and writes log events to a new file.
- **bufferedIO:** Defines the buffer of the log event.
  - **True:** Write access is via buffer.
    - **bufferSize:** Specifies the size of the buffer.
    - **ImmediateFlush:**
      - **True:** Flush after every log message.
      - **False:** No flush after every log message.
  - **False:** The appender deletes existing log files and writes log events to a new file.
- **Log files:** General Logging Information

In the default system setup, all log files are located in the Tomcat log directory (<<CATALINA\_HOME>>/logs). Each day a new log file is created in this directory and the old log file is renamed to <<logfile>>.yyyy-MM-dd.log, the yyyy-MM-dd corresponding to the date.

The file icw-console.log contains the log information from all components.

## 3.6 Audit logs

Audit-specific information is logged to the file `icw-audit.log`. The file should be sorted by time stamp column. This allows the sequence of log events to be reconstructed.

MPI, specifically MPI, also sends IHE-ATNA compliant audit events to a syslog server. Currently, settings can only be made by changing the database's PROPERTY\_CONFIGURATION table:

- `atna.audit.syslog.host:` Host name of the syslog server
- `atna.audit.syslog.port:` Port number of the syslog server



- `atna.audit.sourceId`: Identifier for the program that is doing the auditing
- `atna.audit.enterpriseSiteId`: Identifier for the business enterprise where MPI is being operated

Make these changes before restarting the system.

### 3.7 Persistent modifications to log settings

Changes made to the log setup using JMX apply only to the currently running system. Changes will be lost when the system is restarted.

You can make changes permanent by modifying the file:

```
<<CATALINA_BASE>>/webapps/<<webapp-name>>  
/WEB-INF/classes/log4j.xml
```

Make these changes before restarting the system.

### 3.8 Other log files

Tomcat-specific information (loaded web applications, JVM messages, etc.) is written to the corresponding log information via the `Console Loggers` in the file `<<CATALINA_BASE>>/logs/catalina.out`.



#### NOTE

The file `Die Datei <<CATALINA_BASE>>/logs/catalina.out` contains messages generated by Tomcat. The file may therefore be very large.

### 3.9 Bulk import of patient data

The file interface is intended for use in importing large numbers of patients into the system. In order to do this, files need to be created that describe each patient as an HL7 fragment. Each file must contain at least one patient and is limited by the maximum allowable file size. The name of each file may not end in the character string `old_` and must end in the character string `.xml`.

The files that will be imported must be first assembled in a dedicated folder, and the absolute path to this folder must be given to the file transfer interface prior to importing. The file interface processes all of the files sequentially and tries to import the patient data into the system. Files that could be completely processed will be renamed by adding the prefix `old_` and the suffix `.done`.





Any errors that occur during the import will be noted either in the import log or in an error file. The error file is located in the folder that contains the files that are being imported.

### 3.9.1 Configuring the file interface

The file interface is configured and started from the Java Management Console. At any given time only one import operation per file interface may be active. The end of the import operation is recorded in the log file.

The following steps are necessary to configure the file interface:

1. From the Java Management Console, connect to the target system.
2. (Optional) To improve performance, the following protocol instances can be set under the path, `prg-all-logging` as shown:
  - `Audit.priority=warn`
  - `BatchImportStatistics.priority=info`
3. Activate the function to automatically map new patients to index patients under the path `com.icw.epr=>Context.AutoMapping=>prg-all`, as shown:  
`autoMappingService.NewMappingTaskProcessingEnabled=true`
4. Deactivate
  - `MessageEventSender`
  - `incomingMessageEventFileWriter`
  - `outgoingEventDispatcher`
  - `outgoingAuditEventHandler`
  - `outgoingMessageEventFileWriter`
  - `outgoingATNAEventHandler`
5. Fully specify the directory path of the folder, in which the files to be imported can be found, under the path

```
com.icw.epr=>Context.BatchImportService=>prg-all:  
batchImportService.SourcePath=<Verzeichnispfad>
```

Begin the import using the `startImport` operation of the

`com.icw.epr=>Context.BatchImportService=>prg-all=>batchImportService` instance.



---

**NOTE****Abbreviated imports**

The import process can be shortened by around 45 percent by adding a second complete secondary system. This system must be configured in such a way that it uses the same target database as the primary system. Do this by setting the following entries in the database table `property_configurations` to the same values as for the primary system:

```
db.connection.driver_class  
db.connection.password  
db.connection.url  
db.connection.username  
hibernate.dialect
```

---

## 4 Configuring system properties

### 4.1 System properties in the database

In the database, system properties are listed in table 9: Property items that cannot be set over JMX. You cannot configure the system properties unless the database has already been set up.

The [Tables \[page 57\]](#) in this section: Property Configuration Items, Environment Variables, and Files, contain properties and their corresponding values (examples or presets), a description, and a prioritization. The priority indicates which properties can be modified and which should not be modified. You should go through the properties and modify them if necessary. You can use any database client (GUI or command line interface) to make these modifications.



#### CAUTION

Incorrect initialization of the configuration database will endanger system stability and/or render system components or the entire system inoperative.

Meaning of the priorities:

- 1 - The value must be modified.
- 2 - The value can be modified if necessary.
- 3 - The value should only be modified if absolutely necessary.

### 4.2 Tables

The table below contains the database property configuration settings that cannot be changed via JMX.

Name	Default value	Description	Ranking
common.maxUploadSize	10485760	Maximum size (in bytes) of files to be uploaded.	3
connections.proxy.url	proxyurl:0000	Proxy URL, if activated.	1
connections.proxy.use	FALSE	True: proxy is activated. False: proxy is deactivated.	2

Name	Default value	Description	Ranking
default.locale	de_DE	The default language version. Must be interpretable by the Java class java.util.Locale.	1
mpi.pix.updateNotification.URL	http://localhost:8080/pmsstub/exporter	Target URL for PID profile messages regarding newly generated, modified, deleted, and merged index patients. These are needed for IHE PIX Update Notifications.	1
java.naming.provider.url	ldap://<<ldap-host>>:389	JNDI (LDAP) URL	1
java.naming.security.credentials	<<ldap-password>>	JNDI (LDAP) password	1
jms.activemq.connection.url	failover:(tcp://localhost:61616)	URL of the ActiveMQ message broker.	3
jmx.jmxmp.service.port	9997	Port of the service URL of the jmxmp connector	2
ldap.account.login.failed.max.default	4	Default setting if the number of allowed failed logins was not set in the LDAP.	3
ldap.password.increment.days	100	Validity period of a new password (in days).	2
mpi.clearing.assignNew-MappingTask.reason	Mapping task too complex. Needs further re-search.,SSN of new data set does not match to proposed index patient.,Potential duplicate. Please inform source system.,Review and resolve mapping task.,Other	Comma-separated list if reasons for remapping tasks in the clarification list	1

Name	Default value	Description	Ranking
mpi.clearing. assignNewMappingTask. target	Clearing Head,Source system,Follow- Up	Comma-separated list if mappings for remapping tasks in the clarification list	1
mpi.pid.comparison. personProperties	name,address/ streetAddress- Line,address/ city,address/zip- Code,address/ country- Code,birthPlace/ city,birth- Date,gender	Comma-separated list of XPath expressions for attributes of index patients that have to change for the index patient to be considered changed (and for example to send an update message to the MSB).	3
mpi.pid.updateMessage. onKeepMappingWithout- CopyingFields	false	Determines whether an update message is sent to the MSB when the mapping to an index patient is retained even if the data set was changed and none of the changed data is mapped to the index patient.	3
mpi.pix.updateNotification. enabled	true	Determines whether messages will be sent to the target URL defined under mpi.pix.updateNotification.URL. When IHE PIX Update Notifications are used this value must be set to true.	1
mpi.reevaluateNew MappingTaskOnUp- date.enabled	true	New data sets that have already been evaluated will be rescored each time changes are made to existing index patients.	3
hibernate.show_sql	false	If true, SQL commands are issued in the LOG files. This should only be activated when troubleshooting.	3
jmx.jmxmp.logins	admin= <<jmx- password>>	Login for logging in to the JMX interface (in the format <user-name>=<password>)	1

Name	Default value	Description	Ranking
mpi.clearing.updateOnNewPatientEvent.enabled	false	If true, changes to any existing patient data sets will also be accepted when an admission report is received. If false, changing the data set requires a corresponding change message.	2
mpi.patientEdit.defaultCountryCode	DE	Two-digit country code that identifies the language of the MPI interface when a user generates a dataset.	1
search.fuzzy.alphabet	abcdefghijklmnopqrstuvwxyz- zääääääæ- çèéëëìíîïðñóôõ öøùúûüýþÿß	Characters that are replaced or added for the PDQ fuzzy search to find similar patients.	3
search.regexp.otherid	.*[0-9]+.*	Regular expression for other identifiers used in simple searches.	3
search.regexp.state	[^0-9]{2}	Regular expression for the US state used in the simple search.	3
search.regexp.zipCode	[0-9]{5}	Regular expression for postal code used in the simple search.	3
patientValidationService.enabled	false	When true, insufficient data sets will be placed in the recycle bin.	3
patientValidationService.nullifyDirectly	false	If true, insufficient data sets will be deleted immediately, patientValidationService.enabled must also equal "true".	3
patientValidationService.requirePropertyNamees	person/name/ familyName	Comma-separated list of patient properties (in XPath syntax) for ensuring that a patient master data set is valid.	3
atna.audit.syslog.host	localhost	Syslog server where IHE-ATNA compliant audit event messages are sent.	2
atna.audit.syslog.port	514	Port for the syslog server where IHE-ATNA compliant audit event messages are sent.	3

Name	Default value	Description	Ranking
atna.audit.sourceId	ICWMPI	atna.audit.sourceId: Identifier for the auditing program.	3
atna.audit.enterpriseSiteId	ICWMPI	atna.audit.sourceId: Identifier for the auditing company.	1
mpi.indexPatient.removeIfOrphan	true	Determines whether index patients will be deleted from MPI when they are no longer mapped to any in-house patient.	3
mpi.matcher.consumers	10	Number of threads that perform the complete matching procedure for incoming patient registrations. The sequence of patient messages is guaranteed. More threads means heavier CPU load and more requirement for memory, but increases MPI matching logic throughput.	2
mpi.matcher.result.consumers	10	Number of threads to process the results of the matching process. This number should orient itself on mpi.matcher.consumers.	2
mpi.indexPatient.phoneticAlgorithm	Soundex Algorithm	For all name elements, MPI also saves a phonetic representation that can be called up over the IHE PDQ interface. This setting specifies globally the algorithm that will be used. Permitted options include SoundexAlgorithm, RefinedSoundexAlgorithm, MetaphoneAlgorithm, DoubleMetaphoneAlgorithm, PhoneticAlgorithm and NysiisAlgorithm. This value should be set BEFORE operation, because any changes will mean that NO changes will be possible for phonetic representations that have already been saved.	2

Name	Default value	Description	Ranking
search.fuzzy.name.familyName.filter.expectedElements	300	Expected number of different family names in the database. Used for uncertain IHE PDQ queries.	1
search.fuzzy.name.familyName.filter.size	3000	Internal size of a filter, used with uncertain PDQ searches to avoid unnecessary searches. Should be 7-10 times larger than the expected number of elements for which uncertain searches will be performed.	1
search.fuzzy.name.givenName.filter.expectedElements	200	Expected number of different first names in the database. Used for uncertain IHE PDQ queries.	1
search.fuzzy.name.givenName.filter.size	2000	Internal size of a filter, used with uncertain PDQ searches to avoid unnecessary searches. Should be 7-10 times larger than the expected number of elements for which uncertain searches will be performed.	1
search.fuzzy.name.middleName.filter.expectedElements	300	Expected number of different middle names in the database. Used for uncertain IHE PDQ queries.	1
search.fuzzy.name.middleName.filter.size	3000	Internal size of a filter, used with uncertain PDQ searches to avoid unnecessary searches. Should be 7-10 times larger than the expected number of elements for which uncertain searches will be performed.	1



Name	Default value	Description	Ranking
search.fuzzy.name. otherIds.filter. expectedElements	100	Expected number of person IDs in the database. Used for uncertain IHE PDQ queries.	1
search.fuzzy.name. otherIds.filter. size	3000	Internal size of a filter, used with uncertain PDQ searches to avoid unnecessary searches. Should be 7-10 times larger than the expected number of elements for which uncertain searches will be performed.	1

**Table 10: Property configuration settings that cannot be adjusted via JMX**

The following table contains the placeholders used in this document for directories in the file system.

Wildcards	Description	Example
<<ANT_BINDIR>>	Path for the bin directory of the Ant installation	/usr/local/ant/bin
<<ANT_HOME>>	Path for Ant installation	/usr/local/ant/
<<CATALINA_HOME>>	Path for Tomcat installation	/usr/local/webrowsers/apache-tomcat/
<<CATALINA_BASE>>	Path for the Tomcat instance	/usr/local/webrowsers/prg-all
<<JAVA_BINDIR>>	Path for the bin directory of the Java installation	/usr/local/java/bin
<<JAVA_HOME>>	Path for Java installation	/usr/local/java/
<<JRE_DIR>>	Path for the Java Runtime Environment	/usr/local/java/jre/

**Table 11: Environment variables**

The following table contains other placeholders used in this document.

Wildcards	Description	Example
<<db-server>>	Host name of the database server to which the web application connects.	pxs-db.server.com
<<db-port>>	Port of the database server	5321

Wildcards	Description	Example
<<db-schema>>	Database scheme used by the web application.	EPAD
<<db-username>>	ID of the user, with whom a web application is accessing the database.	epa
<<db-password>>	Password of the user accessing the database.	epa
<<db-driver>>	Driver for the database used.	com.mysql. jdbc.Driver or oracle.jdbc.driver. OracleDriver
<<ldap-host>>	Host name of the directory service server (LDAP server).	ldap.server.com
<<ldap-password>>	LDAP password	Password
<<ldap-port>>	Port of the directory service server (LDAP server).	389
<<app-server>>	Host name of the application server on which the web application runs.	app.server.com
<<solution>>	The web application, on which Tomcat is installed.	pxs-2.3
<<webapp-name>>	Context path of the web application that is attached to the web server's URL in order to access the application.	pxs

**Table 12: Files and directories**



## Glossary

Term	Explanation
ATNA	Audit Trail and Node Authentication; IHE profile
Data Sets	A data set is a package of patient data received from an integrated source system. In Master Patient Index, data sets are mapped and referenced over so-called index patients.
Globaler Identifikator (GID)	Additional identifiers for the patient, such as a social security number. These are not relevant in every case.
GUI	Graphic user interface
HL7	Health Level 7
httpd	Hyper Text Transfer Protocol Daemon
IHE	Integrating the Healthcare Enterprise
JDK	Java Development Kit
JMX	Java Management Extensions
MBean	Managed Bean; Java objects representing resources such as programs, services, components or devices. MBeans are used to observe and manage resources.
MPI	Master Patient Index
MSB	Medical Service Bus: The ICW MSB integrated into a hospital's communications server provides the connection between clinical information systems and the other components. Patient data that is being communicated internally in the hospital is transformed (HL7 v2.x to HL7 v3.x), migrated and forwarded securely over MSB to the Master Patient Index and the virtual medical record.
Patient	A person being treated for a medical disorder or disease.
PDQ	Patient Demographics Query
PIX	Patient Identifier Cross-referencing
URL	Uniform Resource Locator; URLs identify and localize a resource over the network protocol used and where the resource is located on computer networks.
VM	Virtual Machine



## List of Figures

Figure 1: Login in JConsole .....	9
Figure 2: JConsole overview .....	10
Figure 3: JConsole MBeans .....	11
Figure 4: MBeans, corrupt HL7 messages .....	14
Figure 5: Automatic rescoring for insufficient data sets .....	24
Figure 6: Blocking strategies .....	29
Figure 7: Scoring .....	35
Figure 8: Source system-specific configuration .....	42
Figure 9: MBean, settings to define missing values .....	47
Figure 10: Automatically merge index patients .....	50



## List of Tables

Table 1: Directory Overview .....	6
Table 2: Summary of MBeans .....	11
Table 3: Mbean: Saving corrupt HL7 messages .....	15
Table 4: Sample expressions .....	30
Table 5: Attribute values .....	31
Table 6: Scoring Parameters .....	40
Table 7: Standard parameter for scoring.givenName .....	43
Table 8: Replacing the parameters in scoring.givenName .....	43
Table 9: Attributes of the Search Expression Converter .....	49
Table 10: Property configuration settings that cannot be adjusted via JMX .....	63
Table 11: Environment variables .....	63
Table 12: Files and directories .....	64