

USER MANUAL

ProfileManager

Version 2.0

Created by: Andreas Hartmann
Modified by: Andreas Hartmann
Saved at: 6. April 2008

Version: 33

Radebeul, 03/31/08

Preamble

This is a legal agreement between you (either an individual person or a company) and Hartmann media solutions Hmedia. You agree to be bound by this agreement by installing, copying or using the ProfileManager software.

If you do not have a valid license for the ProfileManager software, you are not authorized to install, copy or otherwise use the software.

Provided that you have a valid license and you agree with the terms of this agreement, Hartmann media solutions grants you the following rights:

1. You may install as many instances of the ProfileManager software as you want, if this is in accordance with this agreement.
2. You may copy the ProfileManager software or documents that accompany this software in accordance with this agreement.

The information contained in this document and any other document belonging to the ProfileManager software is subject to change without notice and does not contain any liabilities to incur by Hartmann media solutions. Hartmann media solutions takes no charge for any errors that may appear in these documents. Errors and omissions are excepted. Any documentation that accompanies the ProfileManager software is licensed for internal, non-commercial reference purposes only.

No part of the ProfileManager software or any ProfileManager software artefacts may be in any form given to third parties.

Hartmann media solutions reserves all rights not expressly granted in this license.

You may not reverse engineer, decompile or disassemble this software except that you have a written permission by Hartmann media solutions to do so.

This license applies to all updates and supplements that Hartmann media solutions will provide.

Hartmann media solutions may terminate your license if you fail to comply with the terms and conditions of this license. In this case you have to uninstall all your instances of the ProfileManager software and you have to destroy all copies of the software and any of its artefacts.

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

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

© Copyright 2008

Hmedia

Andreas Hartmann media solutions

Ludwig-Richter-Allee 6

01445 Radebeul

mail@hmedia.de

Alle Rechte vorbehalten.

Gedruckt in Europa

All trademarks and registered trademarks used herein are the property of their respective owners.



Processing note

This document was created using OpenOffice.org 2.4. The original document was saved as `ProfileManager Manual.odt`.

Versions

Date/Version	Editor	Changes
01.04.08	Sebastian Lorenz	Initial version
04.04.08	A. Hartmann	Keyboard utility installation, de-installation

Table of Contents

Preamble	iii
Processing note	v
Versions	v
<i>Preface</i>	<i>ix</i>
1 General	ix
2 About this Document	ix
2.1 Symbols and Conventions	ix
2.1.1 Lists	ix
2.1.2 Hints	x
2.1.3 General Information	x
2.1.4 Warnings	x
2.1.5 Links	x
2.1.6 Keyboard commands	x
2.1.7 Console text	x
<i>Chapter 1</i>	
<i>ProfileManager Overview</i>	<i>11</i>
1 iNEWS without the ProfileManager	11
2 iNEWS with ProfileManager	11
<i>Chapter 2</i>	
<i>Installation and Configuration</i>	<i>13</i>
1 ProfileManager Installation Procedure	13
1.1 Java Installation	13
1.2 .NET Framework Installation	13
1.3 ProfileManager Installation	13
1.4 ProfileManager Licensing	13
1.5 Keyboard Utility Installation	14
2 ProfileManager Configuration	15
2.1 Creating access to iNEWS for the ProfileManager	15
2.2 ProfileManager Configuration Files	16

Chapter 3		
<i>Creating User Profiles</i>		23
	1 Setting up the users profile queue	23
	2 Creating a profile from scratch	23
	3 Copying a profile from one user to another	25
Chapter 4		
<i>Troubleshooting</i>		27
	1 Messages after starting the ProfileManager	27
Chapter 5		
<i>Quick Install Guide</i>		29
	1 On all iNEWS servers	29
	2 On an iNEWS workstation	29
	3 Install the ProfileManager program	29
Chapter 6		
<i>Deinstallation Procedure</i>		31
	1 Deinstall one ProfileManager	31
	2 Remove a users profile	31
	3 Remove the ProfileManager completely	31
	3.1 On all iNEWS servers	31
	3.2 On an iNEWS workstation	31
	3.3 On all iNEWS client machines	32

Preface

1 General

A user profile in iNEWS comprises its toolbar layout and keyboard settings. iNEWS itself does not give the user the option to have more than one toolbar layout where she can choose from. With the help of the Hmedia ProfileManager for iNEWS the user can choose between predefined profiles before logging in to iNEWS. If the user changes a profile, e.g. by creating an extra toolbar, these changes will be saved in the iNEWS database.

2 About this Document

The chapters of this manual cover the following topics:

- Chapter 1 gives an Overview of the ProfileManager.
- Chapter 2 explains the installation and configuration steps of the ProfileManager in detail.
- Chapter 3 shows how to create a profile.
- Chapter 4 reports the error messages and some trouble shooting.
- Chapter 5 gives an overview for the installation and setup procedure.

2.1 Symbols and Conventions

2.1.1 Lists

1. Numbered lists are used if the order of the primary list elements is of importance.
 - a. Alphabetic lists are used if the order of the secondary list element is of importance.
- Bullet lists are used if the order of the primary list elements is not important.
 - Outlined Bullets are used if the order of the secondary list elements is not important.

2.1.2 Hints

Hints and extra information can be found in this area.

The hints in the marginal left area help you to work effectively.

2.1.3 General Information

Information like these help you not to miss important facts.

2.1.4 Warnings

WARNINGS LIKE THESE SHOULD PROTECT YOU FROM COMMON PROBLEMS AND FAILURES.

2.1.5 Links

If there is a text link, it will be represented like this:

[Target](#). This link can be clicked with the mouse while reading online (Ctrl +left mouse button).

The following special character will appear behind the link target [△](#). This character leads back to the link source. Target [△](#)

2.1.6 Keyboard commands

If key combinations are described they are connected with a + sign. **Ctrl +D** means pressing the control key and the 'd' key together.

2.1.7 Console text

Sometimes it is necessary to print the output of a console command or the contents of a file in the text. The following is an example of the response of typing the command „java -version“ on the command prompt:

```
h:\>java -version 1
java version "1.5.0_01" Java(TM) 2 Runtime Environment, Standard 2
Edition (build 1.5.0_01-b08) 3
Java HotSpot(TM) Client VM (build 1.5.0_01-b08, mixed mode) 4
```

CHAPTER 1

ProfileManager Overview

1 iNEWS without the ProfileManager

This is a short explanation of how and where iNEWS stores a users' profile information. The explanation is necessary to understand how the ProfileManager works.

Profile information are stored in the iNEWS database. After the user closes the iNEWS client toolbar layouts, user preferences and sessions are sent to the iNEWS server and stored in a special region of the database. When the iNEWS client is started again with the same user, it downloads these information from the database and presents the program window according to the user profile settings.

But the iNEWS client can also be used with a local database only. In this case there is no connection to any iNEWS server. For not loosing user changed settings when closing the iNEWS client, it always stores the user profile into the Windows Registry too. If iNEWS is started with a local database again it will load the user profile from the Windows registry. The iNEWS client can also be forced to load the user profile from the registry even if it is connected to the iNEWS server. Starting the iNEWS client with the program argument `registryprefs` will cause the profile to be loaded from the Windows registry.

2 iNEWS with ProfileManager

The previous chapter has explained that iNEWS stores the profile of a user into the Windows registry. The iNEWS client can also be forced to load the profile of a user from the registry. The ProfileManager basically works with these two iNEWS features. To get a users profile the profile information can be exported from the Windows registry after the user closed the iNEWS client. The profile information can be saved somewhere else and if the same user wants to log in again she can chose one of the previously saved profiles available. After the user chose a profile the ProfileManager writes this profile into the Windows registry and forces iNEWS to start using the profile from the registry.

To store the profile after it has been exported from the Windows registry the ProfileManager uses the iNEWS database itself. The ProfileManager expects the home directory of the user to have a `profiles` queue. Each story in this queue is expected to be a profile. Therefore the stories in this queue are called profile stories.

Stories that contain profile information are saved in the `profiles` queue in the users home directory. These stories are called profile stories.

After opening the ProfileManager the user has to enter its user name first. Then the ProfileManager will open a connection to iNEWS, go into the profiles queue of the given user and present each story as a profile to chose. After the user has chosen a profile the ProfileManager will download this profile, store it in the Windows registry and start iNEWS with the option to load the profile from registry. When the user logs in to iNEWS the layout will be according to the profile. Any changes to the layout are saved in the Windows registry after the user closed the iNEWS client. The ProfileManager is waiting in the background for iNEWS to close. After the profile has been written to the registry the ProfileManager reads the profile out of the registry and updates the profile in the profile queue of the user.

The keyboard information for a profile is not saved in the registry. The keyboard information is written into the profile story by the administrator (see Chapter 3). After the ProfileManager downloaded a given profile and wrote it into the Windows registry the keyboard number is extracted from the profile. This keyboard number must now be set on the iNEWS server for the given user. Therefore the ProfileManager opens an FTP connection to the file system of the iNEWS server and stores the keyboard information on the iNEWS server as a file in a certain directory. A utility program delivered with the ProfileManager is scanning this directory, reads the incoming keyboard information files and sets the given keyboard for the given user.

The keyboard information utility prints error messages and warnings (wrong keyboard numbers, alphanumeric strings instead of a keyboard number) to the system logfile `/var/log/messages`.

As stated before the ProfileManager is waiting in the background for the iNEWS client to close. As soon as the iNEWS client is closed the ProfileManager writes the profile of the user back into the profile story on the iNEWS server. But this only works if the user is logged in once with the same user name on the same PC. If the user opens two instances of iNEWS and logs in with the same user name the ProfileManager automatically disables itself and does not write the profile back to iNEWS.

IF A USER OPENS TWO INSTANCES OF iNEWS THE PROFILEMANAGER DOES NOT WRITE THE PROFILE BACK INTO THE PROFILE STORY.

CHAPTER 2

Installation and Configuration

1 ProfileManager Installation Procedure

Since the iNEWS client can only be run on computers using Microsoft Windows, the ProfileManager also only runs in Microsoft Windows environments. Before installing the ProfileManager make sure Java and .NET are installed on the same machine.

1.1 Java Installation

The ProfileManager needs a Java Runtime Environment (JRE) 6 or higher. The JRE can be downloaded at <http://java.sun.com>.

The command `java` has to be executable from the command line. The version of the installed JRE can be checked with the command `java -version`. The output of this command should look similar to this:

```
...# java -version 1
java version "1.6.0" 2
Java(TM) SE Runtime Environment (build 1.6.0-b105) 3
Java HotSpot(TM) Client VM (build 1.6.0-b105, mixed mode) 4
5
```

1.2 .NET Framework Installation

Make sure you have the .NET Runtime Environment version 2.0 or higher installed on each machine where the ProfileManager is running. The .NET Runtime Environment can be downloaded from <http://www.microsoft.com/downloads>. Details concerning the .NET installation can also be found on this website.

1.3 ProfileManager Installation

The ProfileManager is shipped with an executable installation file. Find the `setup.exe` file on the ProfileManager installation medium and double click it to start the installation process. Follow the instructions on the screen and accept the license agreement at the end of the installation procedure to finish the installation process.

1.4 ProfileManager Licensing

The ProfileManager needs a license to run. If you don't have a valid license contact Hmedia to get one. A license is delivered in a `license.lic` file. This file has to be copied into the ProfileManager installation directo-

ry. The ProfileManager checks this file every time it starts. If no license is installed the program will exit with the following error message.

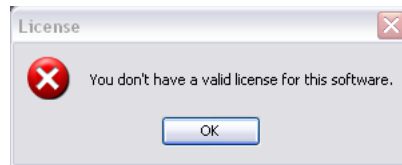


Illustration 1: No-license-message.

1.5 Keyboard Utility Installation

You must install the keyboard utility on all iNEWS servers to enable valid keyboard settings even if the system runs as single. Run the steps as:

1. Copy the `profilemanager-x.x-x.rpm` from the ProfileManager installation medium to any place onto your iNEWS server. `x.x-x` stands for the version number – i.e. `2.0-0`.
2. Switch to user root with `su`.
3. Run `rpm -Uhv <path-to-rpm-file>/profilemanager-x.x-x.rpm`.

This command installs and starts the keyboard utility. The program directory is `/site/scripts/keyboard-changer`. Also an entry for roots crontab is created to run the checker script every 20 minutes.

The default keyboard information file directory is `/tmp/keyboard`. You'll need this information during the configuration procedure. In case you want to change this directory edit the both of `checker.sh` (line 16) and `keyboard-changer.sh` (line 14) in the keyboard utility program directory. Set the preferred path in the variable `KEYBOARDINFDIR` and ensure the chosen directory exists and is writeable for the `pmf_login` user (see Paragraph 2).

checker.sh

```
# /bin/sh 1
# 2
# Author: Andreas Hartmann 3
# andreas@hmedia.de 4
# 5
# supervises the keyboard-changer.sh 6
# runs every 20 minutes (via crontab) 7
# once there is no keyboard-changer.sh process a new one is started 8
# also if the keyboard-changer doesn't respond it'll be replaced 9
# 10
```

```

# check www.hmedia.de/wiki for more information 11
# 12
# 13
PROGRAMDIR="/site/scripts/keyboard-changer" 14
KEYBOARDINFDIR="/tmp/keyboard" 15
LOGFILE="/var/log/messages" 16
LOGDATE=$(date | cut -c5-19) 17
... 18
... 19

```

keyboard-changer.sh

```

#!/bin/sh 1
# 2
# Author: Andreas Hartmann 3
# andreas@hmedia.de 4
# 5
# manages keyboards in according to the Hmedia ProfileManager 6
# runs permanently supervised by checker.sh (via crontab) 7
# 8
# check www.hmedia.de/wiki for more information 9
# 10
# 11
PROGRAMDIR="/site/scripts/keyboard-changer" 12
KEYBOARDINFDIR="/tmp/keyboard" 13
LOGFILE=/var/log/messages 14
LOGDATE=$(date | cut -c5-19) 15
... 16
... 17

```

2 ProfileManager Configuration

2.1 Creating access to iNEWS for the ProfileManager

As stated in Chapter 1 the ProfileManager needs access to iNEWS . Database access is needed to store profile stories in the iNEWS database and file access is needed to send the keyboard information file for a user to the iNEWS server.

The profile stories for each user will reside in the users iNEWS home directory in a queue named `profiles`. The ProfileManager needs access only to these profile queues in the iNEWS database. To give the ProfileManager access to these queues create an iNEWS user who is blacklisted and has read and write access to the profile story queues only. Throughout the rest of this documentation this user will be referred to as the iNEWS ProfileManager user with login name `pm_login` and password `pm_pass`.

For storing keyboard information files the ProfileManager needs access to the file system of all servers belonging to your iNEWS system. Read and write access is only needed to the directory where the keyboard information files should be stored. This directory can be freely chosen by the administrator. It may have already been set in the Keyboard Utility Installation in one of the previous installation steps. To give the ProfileManager access to this directory, create a Linux user for the ProfileManager. This user should only have read and write access to the keyboard information

file directory. Throughout the rest of this documentation this user will be referred to as the iNEWS ProfileManager file server user with login name `pmf_login` and password `pmf_pass`.

2.2 ProfileManager Configuration Files

Before you start with the configuration of the ProfileManager make sure you know:

- the host names or IP addresses of the iNEWS servers.
- the name of the iNEWS home directory.
- the name of the iNEWS ProfileManager user and the iNEWS ProfileManager file server user created in 2.2.1.
- the name of the keyboard information file directory that the Keyboard Utility uses.
- the version and the program directory of the iNEWS client.

Hmedia recommends to use an XML editor when changing the configuration files. They offer syntax highlighting which makes editing the configuration file much easier. Some editors also offer an automatic syntax check which should be used before saving the file. Syntax errors in the configuration files can cause the ProfileManager to stop immediately with an error message.

The ProfileManager has two configuration files. In the `inewsconf.xml` file the connection to the iNEWS servers and available iNEWS clients can be configured. In the `pmconf.xml` file ProfileManager specific settings can be set.

The following is an example of the `inewsconf.xml` file.

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.springframework.org/schema/beans
      http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">
<!-- Section 1 -->
  <bean id="inews-a" class="de.hmedia.inews.ftp.FTPServer">
    <constructor-arg value="172.21.10.1"/>
    <constructor-arg value="pm_login"/>
    <constructor-arg value="pm_pass"/>
    <constructor-arg value="0"/>
    <constructor-arg value="1"/>
    <property name="connectionTimeout">
      <value>3000</value>
    </property>
  </bean>

```



```

36 <bean id="inews-b" class="de.hmedia.inews.ftp.FTPServer">
37     <constructor-arg value="172.21.10.2"/>
38     <constructor-arg value="pm_login"/>
39     <constructor-arg value="pm_pass"/>
40     <constructor-arg value="0"/>
41     <constructor-arg value="1"/>
42     <property name="connectionTimeout">
43         <value>3000</value>
44     </property>
45 </bean>
46
47 <!-- Section 2 -->
48 <bean id="inews_mirror" class="de.hmedia.inews.ftp.FTPMirror">
49     <constructor-arg value="inews"/>
50     <constructor-arg>
51         <list>
52             <ref bean="inews-a" />
53             <ref bean="inews-b" />
54         </list>
55     </constructor-arg>
56 </bean>
57
58 <!-- Section 3 -->
59 <bean id="file-inews-a" class="de.hmedia.inews.ftp.FTPServer">
60     <constructor-arg value="172.21.10.1"/>
61     <constructor-arg value="pmf_login"/>
62     <constructor-arg value="pmf_pass"/>
63     <constructor-arg value="0"/>
64     <constructor-arg value="1"/>
65     <property name="port">
66         <value>600</value>
67     </property>
68     <property name="connectionTimeout">
69         <value>3000</value>
70     </property>
71 </bean>
72
73 <bean id="file-inews-b" class="de.hmedia.inews.ftp.FTPServer">
74     <constructor-arg value="172.21.10.2"/>
75     <constructor-arg value="pmf_login"/>
76     <constructor-arg value="pmf_pass"/>
77     <constructor-arg value="0"/>
78     <constructor-arg value="1"/>
79     <property name="port">
80         <value>600</value>
81     </property>
82     <property name="connectionTimeout">
83         <value>3000</value>
84     </property>
85 </bean>
86
87 <!-- Section 4 -->
88 <bean id="inews" class="de.hmedia.inews.INewsSystem" scope="singleton">
89     <constructor-arg value="inews"/>
90     <constructor-arg ref="inews_mirror"/>
91     <property name="homeQueueName">
92         <value>people</value>
93     </property>
94     <property name="fileFTPServer">
95         <map>
96             <entry>
97                 <key>
98                     <value>file-inews-a</value>
99                 </key>
100                <ref bean="file-inews-a"/>
101            </entry>
102            <entry>
103                <key>
104                    <value>file-inews-b</value>
105                </key>
106                <ref bean="file-inews-b"/>
107            </entry>

```

```

        </map> 108
    </property> 109
</bean> 110
111
<!-- Section 5 --> 112
<bean id="iNEWS2.0" class="de.hmedia.inews.client.INewsClient"> 113
    <constructor-arg value="C:\\Programme\\Avid\\iNEWS\\anws.exe"/> 114
    <constructor-arg value="iNEWS 2.0"/> 115
    <constructor-arg value="2.0"/> 116
</bean> 117
</beans> 118
119

```

The file is separated in five sections. Each section is configuring one or more modules of the same class type (e.g. compare the values of the class variable in line 8 and line 19 of section 1). This file configures:

- the database FTP connections to the iNEWS database in section 1,
- a mirror module which collects all FTP connections of one iNEWS system in section 2,
- the file FTP connections to the file system of the iNEWS servers in section 3,
- the iNEWS system itself, which consists of a mirror and file FTP connections in section 4 and
- one or more iNEWS clients in section 5.

Section 1 defines modules of type FTPServer. Each module defines a connection to the database of an iNEWS server. Therefore each server in an iNEWS system must have a representation as such a module. An FTPServer module has the following structure:

```

<bean id="{server_name}" class="de.hmedia.inews.ftp.FTPServer"> 1
    <constructor-arg value="{server_host_name | server_ip}"/> 2
    <constructor-arg value="{username}"/> 3
    <constructor-arg value="{password}"/> 4
    <constructor-arg value="{loadBalance}"/> 5
    <constructor-arg value="{maxConnections}"/> 6
    <property name="connectionTimeout"> 7
        <value>{timeout_in_milliseconds}</value> 8
    </property> 9
</bean> 10

```

The `{server_name}` value of the `id` variable in line 1 can be set to an arbitrary but unique string value. This value will later be used to add this FTPServer module to an FTPMirror module. The `{server_host_name | server_ip}` value in line 2 must contain the IP address or the host name of an iNEWS server. The `{username}` and `{password}` values in lines 3 and 4 have to be replaced by the iNEWS ProfileManager users login name and password. The `{loadBalance}` and `{maxConnections}` values in lines 5 and 6 can be set to 1. Increasing the values does not affect the system. The `{timeout_in_milliseconds}` value in line 8 defines the maxi-

imum time in milliseconds the ProfileManager will wait for a response when trying to establish a connection to this server.

Section 2 defines modules of type FTPMirror. An FTPMirror collects all database FTP connections defined in section 1 that belong to the same iNEWS system. An FTPMirror module has the following structure:

```
<bean id="{mirror_name}" class="de.hmedia.inews.ftp.FTPMirror"> 1
    <constructor-arg value="{mirror_name}"/> 2
    <constructor-arg> 3
        <list> 4
            <ref bean="{server_name}" /> 5
            <ref bean="{server_name}" /> 6
        </list> 7
    </constructor-arg> 8
</bean> 9
```

The {mirror_name} value in lines 1 and 2 can be set to an arbitrary but unique string value. This value will later be used to add this FTPMirror module to an INewsSystem module. The {server_name} values in lines 5 and 6 point to the id values of two FTPServers defined in section 1 of the file. In this way a list of FTPServers belonging to this mirror is created. If an iNEWS database is made up of more than two iNEWS servers these servers have to be added to this list.

Section 3 defines modules of type FTPServer. Each module defines a connection to the file system of an iNEWS server. An FTPServer module has the following structure:

```
<bean id="{server_name}" class="de.hmedia.inews.ftp.FTPServer"> 1
    <constructor-arg value="{server_host_name | server_ip}"/> 2
    <constructor-arg value="{username}"/> 3
    <constructor-arg value="{password}"/> 4
    <constructor-arg value="{loadBalance}"/> 5
    <constructor-arg value="{maxConnections}"/> 6
    <property name="connectionTimeout"> 7
        <value>{timeout_in_milliseconds}</value> 8
    </property> 9
    <property name="port"> 10
        <value>{ftp_port_for_file_ftp}</value> 11
    </property> 12
</bean> 13
```

This module has a very similar structure as the module in section 2. The {server_name} value of the id variable in line 1 will later be used to add this FTPServer module to an INewsSystem module. The difference to the FTPServer in section 2 is that for {username} and {password} the login of the iNEWS ProfileManager file server user has to be given. In line 11 the FTP port to use for file FTP connections must be given.

Section 4 defines modules of type INewsSystem. Each module defines one of your iNEWS systems that is made up of a mirror and file system FTPServers. The INewsSystem module has the following structure:

```
<bean id="{inews_system_name}" class="de.hmedia.inews.INewsSystem" 1
    scope="singleton"> 2
    <constructor-arg value="{inews_system_name}"/> 3
    <constructor-arg ref="{mirror_name}"/> 4
    <property name="homeQueueName"> 5
```

```

        <value>{home_directory_name}</value> 6
    </property> 7
    <property name="fileFTPServer"> 8
        <map> 9
            <entry> 10
                <key> 11
                    <value>{server_name}</value> 12
                </key> 13
                <ref bean="{server_name}"/> 14
            </entry> 15
        </map> 16
    </property> 17
</bean> 18

```

The `{inews_system_name}` value in line 1 and 3 must match the name of your iNEWS system. The `{mirror_name}` must match the id of a mirror defined in section 2 of this configuration file. The `{home_directory_name}` value in line 6 must contain the path to the users home directories of your iNEWS system. If necessary the path is separated by dots. By default the iNEWS home directory is `PEOPLE`.

Lines 8 to 17 define the iNEWS file FTP servers that belong to this iNEWS system. All file FTP servers that belong to this system must be defined in section 3. The file FTP servers are added to a map. Each entry of the map consists of a key containing an arbitrary name for the server (`{server_name}` in line 12) and a reference to an id of an FTPServer module defined in section 3 (line 14). Since an iNEWS system can consist of up to four iNEWS servers you can add up to four entries in this map. The server names can be chosen as A, B, C and D for example.

Section 5 defines modules of type `INewsClient`. An `INewsClient` module represents an iNEWS client program installed on the same system as this ProfileManager is installed on. This module contains information necessary for the ProfileManager to startup the iNEWS client after the user has chosen a profile. An `INewsClient` module has the following structure:

```

<bean id="{client_name}" class="de.hmedia.inews.client.INewsClient"> 1
    <constructor-arg value="{client_startup_file}"/> 2
    <constructor-arg value="{client_display_name}"/> 3
    <constructor-arg value="{client_version}"/> 4
    <constructor-arg> 5
        <set> 6
            <value>command_line_argument_1</value> 7
            <value>command_line_argument_2</value> 8
        </set> 9
    </constructor-arg> 10
</bean> 11

```

The `{client_name}` value of the `id` variable in line 1 can be set to an arbitrary but unique string value. The `{client_startup_file}` value must point to the iNEWS client executable. Double backslashes have to be used as file separators here. The default iNEWS client startup file is located in `C:\Program Files\Avid\iNEWS\anws.exe`. The corresponding entry in the configuration file would be:

```
<constructor-arg value="C:\\Program Files\\Avid\\iNEWS\\anws.exe"/>
```

The `{client_display_name}` value is just a user friendly name that is shown to the user by the ProfileManager if more than one iNEWS client is configured. The `{client_version}` value is important for the ProfileManager to work correctly. Only the first two numbers are needed (e.g. 2.0 or 2.6). Lines 24 to 29 are not required. They are only used if extra command line parameters have to be added when the client is started up.

The following is an example of the `pmconf.xml` file:

```
<?xml version="1.0" encoding="UTF-8"?> 1
<beans xmlns="http://www.springframework.org/schema/beans" 2
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" 3
      xsi:schemaLocation="http://www.springframework.org/schema/beans 4
                          http://www.springframework.org/schema/beans/spring-beans-2.5.xsd"> 5
 6
  <bean id="inewsContext" class="de.hmedia.inews.INewsContext"> 7
    <constructor-arg value="inewsconf.xml"/> 8
  </bean> 9
10
  <bean id="profileManager" class="de.hmedia.profilemanager.ProfileManager"> 11
    <constructor-arg ref="inewsContext"/> 12
    <property name="keyboardDirectory"> 13
      <value>/tmp/keyboard</value> 14
    </property> 15
    <property name="profileQueueName"> 16
      <value>profiles</value> 17
    </property> 18
  </bean> 19
20
  <bean id="profileManagerModel" class="de.hmedia.profilemanager.client.Pro- 21
fileManagerModel"> 22
    <constructor-arg ref="profileManager"/> 23
  </bean> 24
25
  <bean id="profileManagerFrame" class="de.hmedia.profilemanager.client.Pro- 26
fileManagerFrame"> 27
    <constructor-arg ref="profileManagerModel"/> 28
  </bean> 29
30
</beans> 31
```

In this file only two lines are important for administrators.

In line 14 the directory where keyboard information files have to be stored can be changed. The directory given here is the same directory that was given to the Keyboard Utility in step 1.5 of the ProfileManager installation.

Line 17 contains the name of the profiles queue. This queue is normally set to `profiles` and should not be changed.

CHAPTER 3

Creating User Profiles

1 Setting up the users profile queue

The ProfileManager stores user profiles in profile stories. These stories are located in the profiles queue of the user. Before a profile can be created for a user the user must have a profiles queue. By default the ProfileManager expects the profiles queue to be in the home directory of the user. If the home directory for all users is PEOPLE then the profile stories of a user are expected to be in a queue named:

```
PEOPLE.[initial_of_username].[username].profiles
```

Normally there is only a MAIL and a NOTES queue in the users home directory. The profiles queue has to be added by the administrator manually. By default this queue is named PROFILES.

After creating the profiles queue for a user make sure that the iNEWS ProfileManager user has read and write access to this queue. Otherwise the ProfileManager program will not work.

To store keyboard information in a profile story the ProfileManager uses a story form field. This story form field must be of type `keyboard`. Therefore a form must be assigned to the profiles queue containing a keyboard field. If there is no such field or the keyboard field of a profile story is empty the keyboard will not be set.

The ProfileManager is delivered with a story form and a queue form that can be used for profiles queues. The story form is named PROFILE-S and the queue form is named PROFILE-Q. These forms can be found in the `forms.dump` file on the installation medium. Save this file on the iNEWS server e.g. in the `/tmp` directory. The forms can then be restored by typing the following line to the iNEWS server command prompt:

```
dbrestore d -f /tmp/forms.dump
```

Restoring the dump will add both forms in your forms directory. Assign these forms to the profiles queue of each user.

2 Creating a profile from scratch

This chapter describes the procedure of creating a profile for a user. Once a profile has been created it can be copied to other users as described in the next section of this manual. To create a new profile you need:

- access to the users profile directory,

- Superuser access to iNEWS
- access to the Windows registry and the Windows Registry Editor,
- a text editor like Notepad.

The following steps explain how to create the profile. It is assumed that the user name of the user this profile is created for is *testuser*.

1. If the profile already exists in iNEWS then open the Manage User Accounts dialog in iNEWS and take it as a template and copy it to *testuser*. Then close the iNEWS client.
2. Open the Windows Registry Editor and delete the iNEWS registry key:
3. HKEY_CURRENT_USER\Software\Avid Technology\iNEWS
4. Open iNEWS and login as user *testuser*. Modifications to the profile can be done now.
5. Close the iNEWS client. The profile is now written into the Windows registry.
6. Open the Windows Registry Editor and export the iNEWS registry key:
7. HKEY_CURRENT_USER\Software\Avid Technology\iNEWS
8. Open the exported file with Notepad. Select the whole text and copy it to the clipboard by pressing CTRL+C.
9. Open the iNEWS client, login as a superuser and goto the profiles queue of *testuser*. The profiles queue must be located in the home directory of the user. If *testuser* does not have a profiles queue create one. To know more about creating profiles queues refer to the previous chapter.
10. Create a new story in the profiles queue of *testuser* and give it a name that describes the profile. Paste the clipboard contents into this story's text body by pressing CTRL+V.
11. A keyboard can be assigned to the profile by entering the keyboard number into the keyboard field.

THE PROFILEMANAGER ONLY ACCEPTS THE KEYBOARD NUMBER IN THE KEYBOARD FIELD. THE KEYBOARD NUMBER MUST BE BETWEEN 0 – 255.

12. Save the story and close the iNEWS client.

If you open the ProfileManager now and enter the login name of *testuser* the ProfileManager will present the profiles available for *testuser*.

3 Copying a profile from one user to another

To copy a profile from one user to another just copy the profile story from the profiles queue of the source user to the profiles queue of the target user. Since the profile story contains the user name in the story text, the user name of the source user must be replaced by the user name of the target user. If you are working in an iNEWS 2.6 or higher environment then user names may be suffixed by 'u'. Therefore a user with name *testuser* will appear in the profile story as *testuser* as well as *testuseru*.

CHAPTER 4

Troubleshooting

1 Messages after starting the ProfileManager

Error Message	Solution
<p>After starting the ProfileManager the following message appears: “You don't have a valid license for this software”</p>	<p>The ProfileManager can not find the license.lic file in the ProfileManager program directory or the license you have is not valid. Contact Hmedia to get a valid license.</p>
<p>After opening the ProfileManager or clicking the <i>Reload</i> button the following message appears: “Profile could not be loaded from system. Server is down: ...”</p>	<p>The ProfileManager could not connect to the iNEWS database server. Check the IP addresses or host names of the servers in section 1 of the inewsconf.xml file. For details about this file refer to chapter 2.2.2 of this manual. Try to ping the servers from the machine where the ProfileManager is running on.</p>
<p>After opening the ProfileManager or clicking the <i>Reload</i> button the following message appears: “Profile could not be loaded from system. 530 Logging failed for <i>username</i>.”</p>	<p>The ProfileManager could connect to the iNEWS database server but could not login with the given user name. Check the user names and passwords in section 1 of the inewsconf.xml file. For details about this file refer to chapter 2.2.2 of this manual.</p>
<p>After opening the ProfileManager or clicking the <i>Reload</i> button the following message appears: “Profile could not be loaded from system. No such queue or directory [inews]people.t.testuser.profiles.”</p>	<p>The ProfileManager could not find the profiles queue of the given user (in this case the user login name is <i>testuser</i>). Check if the given path exists. There are three possible reasons for this failure. Check them in the following order:</p> <ol style="list-style-type: none"> 13. The iNEWS home directory is not set correctly. The home directory on an iNEWS system can be set in section 4 of the <code>inewsconf.xml</code> file. For details about this file refer to chapter 2.2.2 of this manual. 14. The given user does not exist. 15. The user does not have a profiles queue. 16. The profiles queue in iNEWS has a different name than stated in the <code>pmconf.xml</code> file. For details about this file refer to chapter 2.2.2 of this manual.
<p>After clicking OK in the ProfileManager to login with the chosen profile the following message appears: “Keyboard could not be set on server. Server is down: 192.168.1.2”</p>	<p>The ProfileManager could not connect to the iNEWS server to store the keyboard information file. Check the IP addresses or host names given in section 3 of the <code>inewsconf.xml</code> file. For details about this file refer to chapter 2.2.2 of this manual. Try to</p>

	ping the servers from the machine where the ProfileManager is running on.
After clicking OK in the ProfileManager to login with the chosen profile the following message appears: “Keyboard could not be set on server. 530 Login incorrect.”	To store the keyboard information file the ProfileManager could connect to the iNEWS server but could not login with the given user name. Check the user names given in section 3 of the <code>inewsconf.xml</code> file. For details about this file refer to chapter 2.2.2 of this manual.
After starting the ProfileManager an error dialog appears immediately showing an error text message starting with: “ProfileManager configuration could not be read. ...”	One of the ProfileManager configuration files (<code>inewsconf.xml</code> or <code>pmconf.xml</code>) is not correct. The error messages are printed in reverse order as they appeared in the system. Follow the error messages from the last line to the second to track the error. If the last line states some kind of syntax error check the syntax of the configuration file that has been altered at last. If it is not a syntax error the message will explain which module in the configuration file is not correct.

CHAPTER 5

Quick Install Guide

1 On all iNEWS servers

1. Install the ProfileManager Keyboard Utility.
2. Optional: Set the keyboard information file directory in the Keyboard Utility.
3. Create the iNEWS ProfileManager file server user who has read and write access to the keyboard information file directory.
4. dbrestore the forms.dump into your database.

2 On an iNEWS workstation

1. Create the iNEWS ProfileManager user that will have read and write access to the profiles queues of all users.
2. Create a profiles queue for a user.
3. Give the profiles queue the story form PROFILE-S and queue form PROFILE-Q.
4. Create a profile for the user. Refer to chapters 3.2 and 3.3 for more information.
5. Repeat steps 2 – 4 for all users that should have a profile.

3 Install the ProfileManager program

1. Run the ProfileManager installation file.
2. Copy the license into the ProfileManager program directory.
3. Update the inewsconf.xml and if necessary the pmconf.xml configuration file.

CHAPTER 6

Deinstallation Procedure

1 Deinstall one ProfileManager

To remove a ProfileManager program off an iNEWS workstation use the Uninstall tool delivered with Microsoft Windows (Control Panel – Software)

2 Remove a users profile

To wipe out one or more profiles for a given user simple kill the relevant profile stories from this users profiles queue.

To remove any profile for that user also remove the profiles queue of this particular user.

3 Remove the ProfileManager completely

3.1 On all iNEWS servers

1. Uninstall the ProfileManager Keyboard Utility with running
`rpm -ev profilemanager-x.x-x` as root. `x.x-x` stands for the version number. This stops and removes the program completely including the crontab entry.
2. If you have customised the keyboard information file directory during the installation, remove that directory manually. The default directory `/tmp/keyboard` is removed automatically by the rpm uninstall in step 1.
3. Eventual delete the iNEWS ProfileManager file server user.

3.2 On an iNEWS workstation

1. Delete the iNEWS ProfileManager user.
2. Remove all profiles queues in the users home directories.
3. Delete the `system.forms.p.profile-s` and `system.forms.p.profile-q` queues.
4. Give all users their appropriate profiles in the iNEWS user manager dialog

3.3 On all iNEWS client machines

Delete the ProfileManager using the Microsoft Windows uninstall tool.