

Installation Manual

RAID:FMS - Fraud Management Solution



Revision History

Version	References	Author	Issue Date	Comments
1.0		WeDo Technologies	07-05-2015	RAID FMS 8.0



This document belongs to WeDo technologies and all information included is strictly confidential all rights reserved, no part of this publication may be produced, stored in a retrieval system, or transmitted in any form or by any means electronic, mechanical photocopying or otherwise.

Contents

1	Introduction.....	7
1.1	GOALS.....	7
1.2	SCOPE	7
1.3	AUDIENCE	7
1.4	REFERENCES.....	7
1.5	DEFINITIONS	7
2	General Overview	8
2.1	PRODUCT ARCHITECTURE	8
2.1.1	RAID:FMS Backoffice Server.....	9
2.1.2	Portal Server	10
2.1.3	Frontoffice Web Application.....	10
2.1.4	Backoffice Admin Application.....	10
2.1.5	Licensing in RAID FMS	10
2.2	PLANNING YOUR INSTALLATION	11
2.3	INSTALLATION MODE	12
3	Pre-Installation Requirements.....	13
3.1	HARDWARE	13
3.2	SOFTWARE.....	13
3.2.1	Server.....	13
3.2.2	Windows Client	14
3.2.3	Web Client	14
3.3	DATABASE INSTALLATION REQUIREMENTS.....	15
3.3.1	Oracle Side Installation Full	16
3.3.2	Oracle Side Installation Partial.....	17
3.3.2.1	Creation of <useradm>_SysPriv Role	17
3.3.2.2	Creation of <userapp>_SysPriv Role	18
3.3.2.3	Creation of <useradm> User	18
3.3.2.4	Creation of <userapp> User	18
3.3.2.5	Creating <userdat> User.....	18
3.3.2.6	Definition of Quota Unlimited	19
3.3.3	Oracle Side Manual Installation	19
4	Installing RAID:FMS Server	21



4.1	SOFTWARE DEPLOYMENT	22
4.2	SETUP INSTANCE CONFIGURATION FILE.....	22
4.2.1	General Parameters.....	25
4.2.2	Database Parameters	30
4.2.2.1	Scheduler component.....	34
4.2.3	Server Parameters	34
4.2.4	Auditing Parameters	36
4.2.5	Portal Integration	37
4.2.6	Change Server Parameters after Installation.....	39
4.2.7	Protecting passwords	40
4.3	CREATE AND RUN RAID:FMS INSTANCE SERVER	40
4.3.1	Run RAID:FMS Instance Server	41
4.4	UPGRADING THE RAID:FMS SERVER.....	42
4.5	UNINSTALL SERVER	43
4.5.1	Uninstall Process.....	43
4.5.2	Installation Error	44
4.5.3	Reusing DAT Schema in Upgrade	44
4.6	SYSTEM PROPERTIES CONFIGURATION	44
4.6.1	General Properties	45
4.7.1	Basics	45
4.7.1.1	Key store	45
4.7.1.2	Certificate	45
4.7.2	Application server	46
4.7.3	Application Browser.....	46
5	Auditing Features	48
5.1	AUDITING LOG.....	48
5.1.1	Audit Logger File	48
5.1.2	Filter File.....	50
5.1.2.1	Base Definition	51
5.1.2.2	System Events	51
5.1.2.3	Operation Filter	52
5.1.2.4	Filter Sample	52
6	Installing a RAID:FMS Satellite Server.....	54
6.1	SETTING UP THE INSTANCE CONFIGURATION FILE.....	54

6.2	CREATING AN INSTANCE OF A RAID:FMS SATELLITE SERVER.....	55
7	Deploying Application Portlets	56
8	How to Install an Add-on.....	57
9	How to Install GUIs	59
10	Troubleshooting	64
10.2	ERROR STARTING COMPONENT SEARCH	64
10.3	ERROR STARTING COMPONENT IM.CORE	65
10.4	ERROR STARTING COMPONENT EVTTS.....	66
10.5	EVTTS LIBRARY	66
10.6	ERROR RENDERING WEB CONTENT	67
11	Appendix	68
11.1	RECOVERING A FAILED INSTALLATION	68
11.1.1	Rolling Back Installation	68
11.1.1.1	Full Database Recovery	69
11.1.1.2	Table-based Recovery.....	70
11.2	BACKUP	70
11.2.1	Database Backup.....	70
11.2.2	Filesystem Backup	72

List of Tables

Table 1 – Minimum versions of GCC that can be used in each platform/operating system	14
Table 2 - Additional options for software deployment command.....	22
Table 3 - config-generator command options.....	23
Table 4 - apply-instance-config command options	39
Table 5 - encrypt-text Command Parameters	40
Table 6 - create-instance Command options.....	42
Table 7 - update-instance Command options.....	43
Table 8 - drop-instance Command Parameters and execution example	43
Table 9 - Audit Log Line fields	49
Table 10 - Operation Relevant fields	51
Table 11 - List of Add-ons.....	57

List of Figures

Figure 1 – Raid:FMS architecture telecom	8
Figure 2 - Product architecture	9
Figure 3 – RAID:FMS Backoffice server	9
Figure 4 – Installation process	12
Figure 5 – Server Installation phases	21
Figure 6 – RAID:FMS uninstall	59
Figure 7 – Select components to uninstall	60
Figure 8 – Uninstall process conclusion	60
Figure 9 – RAID:FMS installation	61
Figure 10 – License agreement terms	61
Figure 11 – Selection of the components to install	62
Figure 12 – Installation directory	62
Figure 13 – Installation process conclusion	63
Figure 14 – Web client side rendering error	67
Figure 15 – Table-based recovery	70
Figure 16 – Full point-in-time recovery	71

1 Introduction

1.1 Goals

This document provides information about the installation procedure for the RAID:FMS (Fraud Management) product. This is a technical document which presumes some prior knowledge of system administration.





1.2 Scope

This document is only intended for product installation of RAID:FMS. The portal installation component, which is required for this installation, is detailed in an external document. The portal needs to be installed prior to this installation/upgrade.

1.3 Audience

Due to its technical characteristics, this manual is written for consultants or technical users responsible for the RAID:FMS installation/operation

1.4 References

-  [OMF80] OMN_RAID_001_E - RAID FMS 8.0 - Operation Manual
-  [MG80] DOC_RAID_001_E - RAID FMS 8.0 - Configurations Migration Guide
-  [PRT16] IMN_WEDOPORTAL_001_E - WP 1.6 - Installation Manual
-  [CM34] IMN_ACTIVIS_001_E - CM 3.4 - Installation Manual

1.5 Definitions

AF	WeDo Technologies' Application Framework
BCM	Business Concepts Manager
BPM	Business Process Manager
CMF	Connection Manager Factory
CSM	Context Search Manager
EH	Event Handler
FC	Fraud Center
GUI	Graphical User Interface
IM	Integration Module
IUD	Insert Update Delete tasks
JVM	Java Virtual Machine
SSL	Secure Sockets Layer

2 General Overview

RAID:FMS is a Fraud Management System that can be used either as a stand-alone solution or it can be integrated with the Revenue Assurance (RAID) solution.

The following figure shows the RAID:FMS architecture:

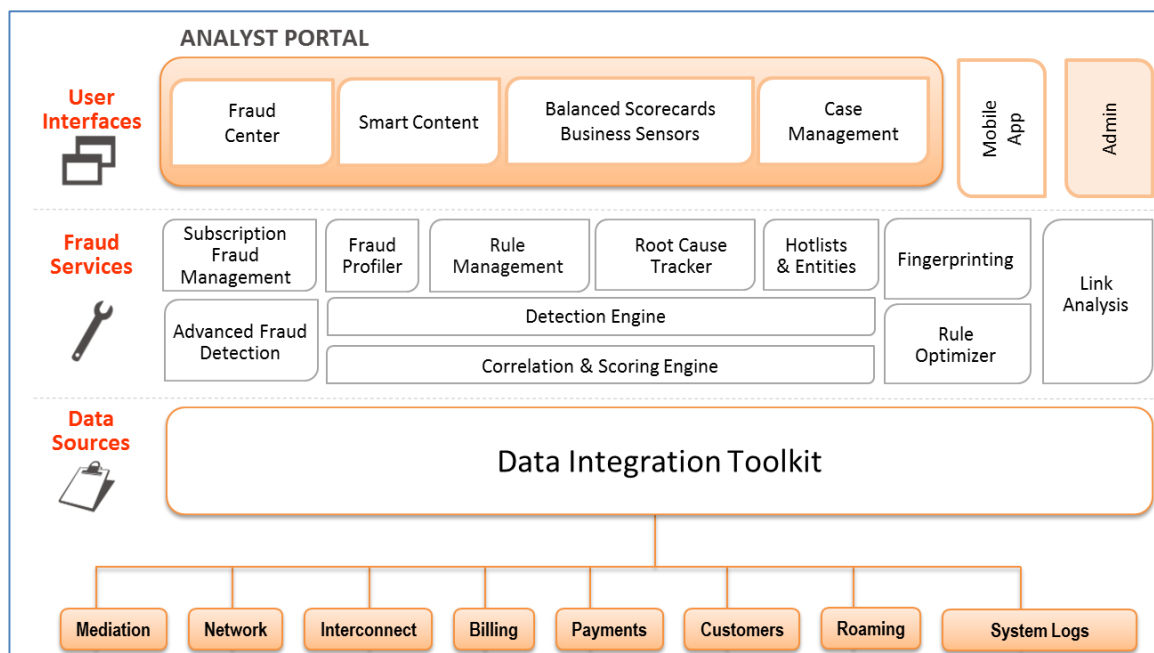


Figure 1 – Raid:FMS architecture telecom

2.1 Product Architecture



RAID:FMS is integrated in a unified user interface which can be shared with other WeDo products, referred to in this document as "Portal". The "Portal" needs to be installed prior to this installation. During installation, each module registers itself in the portal database and some artifacts are deployed in the portal server.

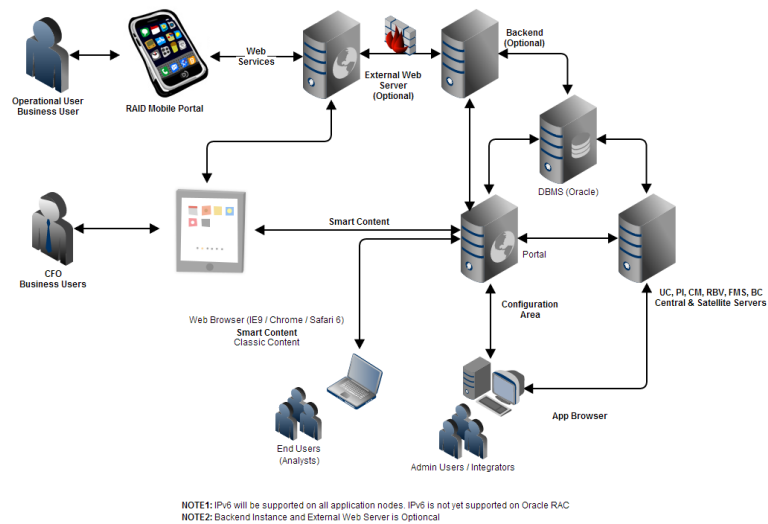


Figure 2 - Product architecture

The RAID:FMS product architecture is composed of four distinct nodes:

- RAID:FMS Backoffice Server;
- Portal Server;
- Frontoffice Web Application;
- Backoffice Admin Application.

The following sections provide a brief introduction to each node.

2.1.1 RAID:FMS Backoffice Server

The main RAID:FMS server instance – **Fraud Center** – is the primary engine for executing server-side business logic, providing services to remote client machines or applications like the Portal Server or the Backoffice Admin Application.

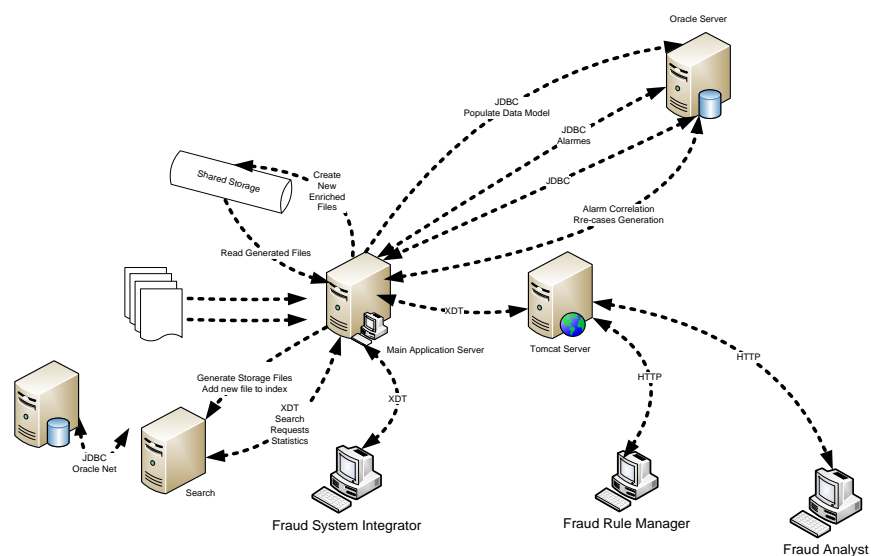


Figure 3 – RAID:FMS Backoffice server

In order to improve performance, additional servers may be installed – like **Satellite Servers**. The RAID:FMS Satellite Servers have the purpose of supporting the Fraud main server, offering additional processing power. The Satellite servers share the Main Server configuration; no additional database objects are created.

There is also the possibility of adding new modules to an existing Main Server, such as the Business Control module. These supplements are distributed as add-ons, which must be installed over an existing server, as described in chapter 7.

The installation steps for the RAID:FMS Main Server are detailed in chapter 4 and the installation steps for a RAID:FMS Satellite Server are detailed in chapter 6.

2.1.2 Portal Server

The Portal server installation is beyond the scope of this manual (see [\[PRT16\]](#)). RAID:FMS installation produces web applications/portlets which can be deployed in the portal.

If configured so, RAID:FMS portlets and applications are auto-deployed during the installation.

2.1.3 Frontoffice Web Application

This is a portlet-based application where users can perform environment customizations. The RAID:FMS data is available inside specific RAID:FMS portlets and by using the generic Data Listing and Perspective portlets. From the web application you can, for instance, access your working inbox, manage fraud cases, extract reports, perform queries, etc.

Installation steps for the RAID:FMS Web Application can be found in chapter 7.

2.1.4 Backoffice Admin Application

The Windows Client is a part of the RAID:FMS module that allows the user to interact with and use the features provided by the server. This is mainly used by system integrators and system administrators.

Installation steps for RAID:FMS Windows Client can be found in chapter 9.

2.1.5 Licensing in RAID FMS

RAID:FMS is delivered with four modules:

- The Core module, essential for the main operations of fraud detection:
 - Loading, Detection, Correlation, Hotlists, Entity Exemption, Scoring, Profiling, Case Management and Data Visualization.
- Earlier Detection module (ED), contains the preventive features of a RAID-FRAUD solution:
 - Subscription Fraud, Fingerprinting and Advanced Fraud Detection.

- Deeper Investigation module (DI), contains the features that allow a deeper investigation on the cause of the results previously obtained:
 - Events Tracking and Link Analysis.
- Tune System module (TS), contains the features that makes possible to tune rules of the detection engines:
 - Rules Optimizer.

The Core module is always installed, independently of the policies defined in the license.

The other 3 modules are installed if the license grants enables their installation and are independent between each other.

If a module is not installed, it will not be possible to use the included features: the features will not be available.

When installing/upgrading RAID:FMS version 8.0, you should have an adequate license to enable the features. Please consult the WeDo License Support Desk to obtain the necessary information and the license file.

2.2 Planning your Installation

The RAID:FMS installation process includes the following steps:

1. Pre-Installation Requirements - This chapter describes pre-installation tasks that you must complete before installing the product;
2. Software Deployment - Follow the steps described in this section to unpack the project package;
3. Setup Instance Configuration File - This section describes the configuration file for RAID:FMS server instance and its properties;
4. Create and Run RAID:FMS Instance Server - This section refers to the creation steps for a server instance;
5. Installing a RAID:FMS Satellite Server - This chapter refers to the creation steps for a satellite server instance;
6. How to Install an Add-on - This chapter refers to the steps required to properly deploy and install an add-on;
7. How to Install GUIs - This chapter explains how to install the RAID:FMS Graphical User Interfaces;
8. Deploying Application Portlets - Follow these steps to setup and install RAID:FMS Web Client.

The diagram below offers a visual representation of the overall installation process.

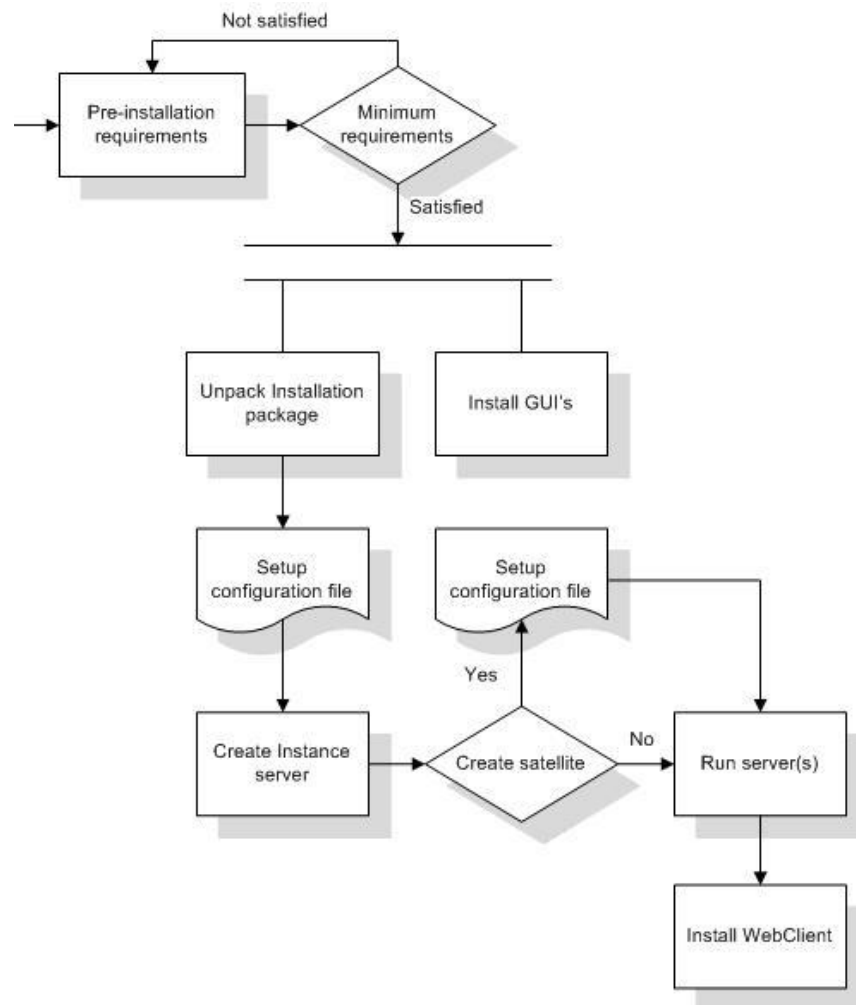


Figure 4 – Installation process

2.3 Installation Mode

RAID:FMS provides three installation types: **Basic**, **Intermediate** and **Advanced**. Each installation mode is to be used according to the customer's needs and requirements.

- **Basic:** In this installation mode, the installation engine is responsible for performing all tasks related to the Oracle installation side. The engine creates Oracle users, roles and objects;
- **Intermediate:** Database users and roles are created in advance by the customer's DBA team. The RAID:FMS installation is only responsible for creating necessary data tables and inserting respective data;
- **Advanced:** In this installation mode, a DDL file is created containing all statements responsible for creating Oracle users, roles and objects.

3 Pre-Installation Requirements

3.1 Hardware

The necessary hardware for RAID:FMS installation is dependent on its configuration. The hardware sizing must therefore be specified for each particular scenario, depending on the implementation scope.

3.2 Software

This chapter describes necessary software requirements. The chapter is divided into three sections representing the software requirements for the RAID:FMS Server installation, the Windows Client installation and the Web client installation.

3.2.1 Server

RAID is supported on the following architectures:

- HP-UX 11.31
 - Itanium - IA64
- Oracle Solaris 11.2
 - Intel/AMD X86-64
 - SPARC 64
- Red Hat Enterprise Linux 7.x, Cent OS 7.x, Oracle Enterprise Linux 7.x
 - X86-64
- IBM AIX 7.1
 - PPC64

The software needed for server installation/runtime is as follows:

- Java Runtime Environment **7.0u80**, **8.0u45** or greater (an up-to-date version in the same major version with the latest patches is recommended); only SUN™, HP™ and IBM™ VMs are certified to work with the product. When using RBV features, the full JDK is still needed.
- Oracle 12c, Oracle 11g DBMS
 - SQLLoader tool must be available in the environment when xDR archiving tools are used
- GCC/GCC-C++/LIBSTDC++ - According to the table below.
- WeDo Portal 1.6.x (The WeDo Portal installation where the RAID portlets is deployed).


The minimum disk space needed for server installation is approximately 4096 MB. Additional space is necessary to store log, data and other files to ensure full operation of the RAID server.

The minimum recommended memory size to run each server instance is 2048 MB. This value can be modified depending on each implementation parameter configuration.

The following table lists the minimum versions of GCC that can be used in each platform/operating system. The compilers and associated runtimes can be obtained for each OS provider as a bundle (depot/rpm) in their usual locations.

Operating System / Versions	Minimum GCC/G++/libstdc++
CentOS Linux 7.x – X86-64	4.8.3
Redhat Enterprise Linux 7.x – X86-64	
Oracle Enterprise Linux 7.x – X86-64	
Oracle Solaris 11.2 – SPARC	3.4.3
Oracle Solaris 11.2 – X86-64	3.4.3
HP-UX 11.31 – IA64	4.2.3
IBM AIX 7.1 – PPC64	4.2.0

Table 1 – Minimum versions of GCC that can be used in each platform/operating system


	The product is able to work without an installed C compiler but some features used on the xDR search component will be slower.
---	--

3.2.2 Windows Client

The RAID:FMS Windows Client can be installed in the following operating systems:

- Microsoft Windows 7 (32 & 64 bits);
- Microsoft Windows 8 (64-bits);
- Microsoft Windows 8.1 (64-bits).

The minimum disk space needed for client installation is approximately 512 MB. The minimum memory size to run the client is 1024 MB (32-bit) and 2048 MB (64-bit).

	The embedded web interface used for some features inside Application Browser will not work properly if Internet Explorer 10 or 11 is not available.
---	---

3.2.3 Web Client


The RAID:FMS web client application embedded in the unified application portal can be accessed by the following browsers and related operating systems:

Device	OS/Browser	Chrome	Firefox	IE	Safari
Desktop	Windows 7, 8, 8.1 (RT not supported)	>= 40	>= 24 ESR >= 32	10 11	
Desktop	Mac OSX (Lion, Mountain Lion, Mavericks, Yosemite)	>= 40	>= 24 ESR >= 32		6
Tablet	IOS (8 or greater)				(built-in)
Tablet	Android (4.0 or greater)	>= 40			

Some graphics are presented as Java Applets (when classic contents are used), which require a Java 8.0 > u45 JRE or greater running in the client's machines. These charts/diagrams are not supported for tablets and will be discontinued on chrome during 2015 (see note below).

On desktop web browsers and for very complex pages, it's advisable to have at least 4GB of memory (bare minimum is 2GB). Please note that this requirement may vary due to project implementation aspects.

On Internet Explorer, the compatibility mode must be disabled for the product site. If this is set, product will not render properly (see IE Settings for details).

	Be aware that support for applets in chrome will be discontinued during 2015 (http://blog.chromium.org/2014/11/the-final-countdown-for-npapi.html). If this is still needed, please apply the temporary workarounds or use a different web browser (e.g. Firefox, Safari, IE)
---	---

3.3 Database Installation Requirements

To prepare the Oracle environment to setup the database side of RAID:FMS installation, three methods can be used. The selection of the method to be applied depends entirely on the client demands.


- **RAID:FMS – Oracle Side Installation Full** (Users and Roles created using create-instance script);
- **RAID:FMS – Oracle Side Installation Partial** (No Users or Roles created by create-instance script);
- **RAID:FMS – Oracle Side Manual Installation** (No database objects are created).

This chapter introduces these three methods.

Note: If the RAID:FMS installation is to hold non-English data, the following database configurations must be taken into consideration:

- The `NLS_CHARACTERSET` parameter should be set to a character set that supports the intended data. The `AL32UTF8` (or others) can be used when support for any Unicode character is required.
- For multi-byte character sets (such as `UTF8`), the `NLS_LENGTH_SEMANTICS` parameter should be set to `CHAR`, instead of the default `BYTE`.

Since Oracle 11, the default password policy forces passwords to be changed regularly. In order to ensure your processes continue to run properly, you need to change the default profile to disable password expiration and account lockout. The following settings on the default profile are recommended:

	<code>FAILED_LOGIN_ATTEMPTS</code>	10
	<code>PASSWORD_LIFE_TIME</code>	UNLIMITED
	<code>PASSWORD_REUSE_TIME</code>	UNLIMITED
	<code>PASSWORD_REUSE_MAX</code>	UNLIMITED
	<code>PASSWORD_LOCK_TIME</code>	UNLIMITED
	<code>PASSWORD GRACE TIME</code>	UNLIMITED

3.3.1 Oracle Side Installation Full

The installation process performs all the tasks related to the Oracle installation side of RAID:FMS automatically. This includes the creation of Oracle users and roles. This is the [basic](#) installation mode.

To perform this installation, a database user (further referred in installation as <DbUsersDbUser>) is required with the following Oracle privileges:

- Grant Create Session With Admin Option;
- Grant Create User;
- Grant Drop User;
- Grant Alter User;
- Grant Create Role;
- Grant Create Type With Admin Option;
- Grant Drop Any Role;
- Grant Create View With Admin Option;
- Grant Query Rewrite With Admin Option;
- Grant Create Materialized View With Admin Option;
- Grant Create Trigger With Admin Option;
- Grant Create Table With Admin Option;

- Grant Create Synonym With Admin Option;
- Grant Create Sequence With Admin Option;
- Grant Create Procedure With Admin Option;
- Grant Create Database Link With Admin Option;
- Quota Unlimited in all tablespaces that RAID:FMS uses in the database.

3.3.2 Oracle Side Installation Partial

This section describes how to manually create the RAID:FMS users. In this case, all users and roles are created by the customer's DBA team, leaving the installation engine in charge of creating the database objects. This corresponds to the [intermediate](#) installation mode.

To perform the installation, a database user (further referred in installation as <DbUsersDbUser>) is required with the following privileges:

- Grant Create Session.

For a successful installation of RAID:FMS with the user indicated above, the customer's DBA team should use an Oracle user that has at least the permissions of section 3.3.1, to execute in sequence the sections 3.3.2.1 thru 3.3.2.6. After that, the installation process only needs to perform the objects creation over these users.

3.3.2.1 Creation of <useradm>_SysPriv Role

The following role must be created and granted to the user <useradm>:

```
CREATE ROLE <useradm>_SysPriv NOT IDENTIFIED;

GRANT SELECT ON SYS.V_$RESERVED_WORDS TO <useradm>_SysPriv;
GRANT CREATE VIEW TO <useradm>_SysPriv;
GRANT CREATE TRIGGER TO <useradm>_SysPriv;
GRANT CREATE TABLE TO <useradm>_SysPriv;
GRANT CREATE SYNONYM TO <useradm>_SysPriv;
GRANT CREATE MATERIALIZED VIEW TO <useradm>_SysPriv;
GRANT CREATE SESSION TO <useradm>_SysPriv;
GRANT CREATE SEQUENCE TO <useradm>_SysPriv;
GRANT CREATE PROCEDURE TO <useradm>_SysPriv;
GRANT CREATE DATABASE LINK TO <useradm>_SysPriv;
GRANT QUERY REWRITE TO <useradm>_SysPriv;
GRANT CREATE TYPE TO <useradm>_SysPriv;
```

3.3.2.2 Creation of <userapp>_SysPriv Role

The following role must be created to be granted to the user <userapp>:

```
CREATE ROLE <userapp>_SysPriv NOT IDENTIFIED;

GRANT SELECT ON SYS.V_$RESERVED_WORDS TO <userapp>_SysPriv;
GRANT CREATE SYNONYM TO <userapp>_SysPriv;
GRANT CREATE SESSION TO <userapp>_SysPriv;
```

3.3.2.3 Creation of <useradm> User

```
CREATE USER <useradm> IDENTIFIED BY <useradmpassword>
  DEFAULT TABLESPACE <smalltablespace>
  TEMPORARY TABLESPACE <temporarytablespace>
  PROFILE DEFAULT
  ACCOUNT UNLOCK;

GRANT <useradm>_SysPriv TO <useradm>;
ALTER USER <useradm> DEFAULT ROLE <useradm>_SysPriv;

CREATE ROLE <useradm>_SIUD NOT IDENTIFIED;
CREATE ROLE <useradm>_S NOT IDENTIFIED;
```

3.3.2.4 Creation of <userapp> User

```
CREATE USER <userapp> IDENTIFIED BY <userapppassword>
  DEFAULT TABLESPACE <smalltablespace>
  TEMPORARY TABLESPACE <temporarytablespace>
  PROFILE DEFAULT
  ACCOUNT UNLOCK;

GRANT <userapp>_SysPriv TO <userapp>;
GRANT <useradm>_SIUD, <useradm>_S TO <userapp>;
ALTER USER <userapp> DEFAULT ROLE <userapp>_SysPriv, <useradm>_SIUD, <useradm>_S;
```

3.3.2.5 Creating <userdat> User

This user is only required if the product tables and configuration tables are stored in different schemas, otherwise the <useradm> is used for both types of tables.

```
CREATE USER <userdat> IDENTIFIED BY <userapppassword>
  DEFAULT TABLESPACE <smalltablespace>
  TEMPORARY TABLESPACE <temporarytablespace>
  PROFILE DEFAULT
  ACCOUNT UNLOCK;
```

```
GRANT <useradm>_SysPriv TO <userdat>;
ALTER USER <userdat> DEFAULT ROLE <useradm>_SysPriv;
```

3.3.2.6 Definition of Quota Unlimited

For the users created before, the following instruction should be executed for all the database tablespaces where users should have access.

```
ALTER USER <user> QUOTA UNLIMITED ON <tablespace>;
```

Where:

- <user> are: <useradm>, <userapp> and <userdat> (if it exists);
- <tablespaces> each tablespace defined on RAID:FMS installation configure file:

```
RAID_FRAUD_SMALL_TABLE_TABLESPACE
RAID_FRAUD_MEDIUM_TABLE_TABLESPACE
RAID_FRAUD_LARGE_TABLE_TABLESPACE
RAID_FRAUD_SMALL_INDEX_TABLESPACE
RAID_FRAUD_MEDIUM_INDEX_TABLESPACE
RAID_FRAUD_LARGE_INDEX_TABLESPACE
RAID_FRAUD_LOB_TABLESPACE
RAID_FRAUD_ADDITIONAL_TABLESPACES
```

3.3.3 Oracle Side Manual Installation

In this case, a DDL file is generated at install time containing the statements that create database users, roles and necessary objects. The procedure to create this file is explained later in section 4.3. This is the [advanced](#) installation mode where the customer may customize the DDL file.

To perform this installation, a user on the database (further referred in installation as <DbUsersDbUser>) is required with the following privileges:

- Grant Create Session With Admin Option;
- Grant Select On Sys.V_\$Reserved_Words With Grant Option;
- Grant Create User;
- Grant Drop User;
- Grant Alter User;
- Grant Create Role;
- Grant Drop Any Role;
- Grant Create View With Admin Option;
- Grant Create Type With Admin Option;
- Grant Query Rewrite With Admin Option;
- Grant Create Materialized View With Admin Option;

- Grant Create Trigger With Admin Option;
- Grant Create Table With Admin Option;
- Grant Create Synonym With Admin Option;
- Grant Create Sequence With Admin Option;
- Grant Create Procedure With Admin Option;
- Grant Create Database Link With Admin Option;
- Quota Unlimited in all tablespaces that RAID:FMS uses in the database.

4 Installing RAID:FMS Server

The server installation contains three phases:

- Unpack installation package;
- Setup configuration file;
- Create instance server.

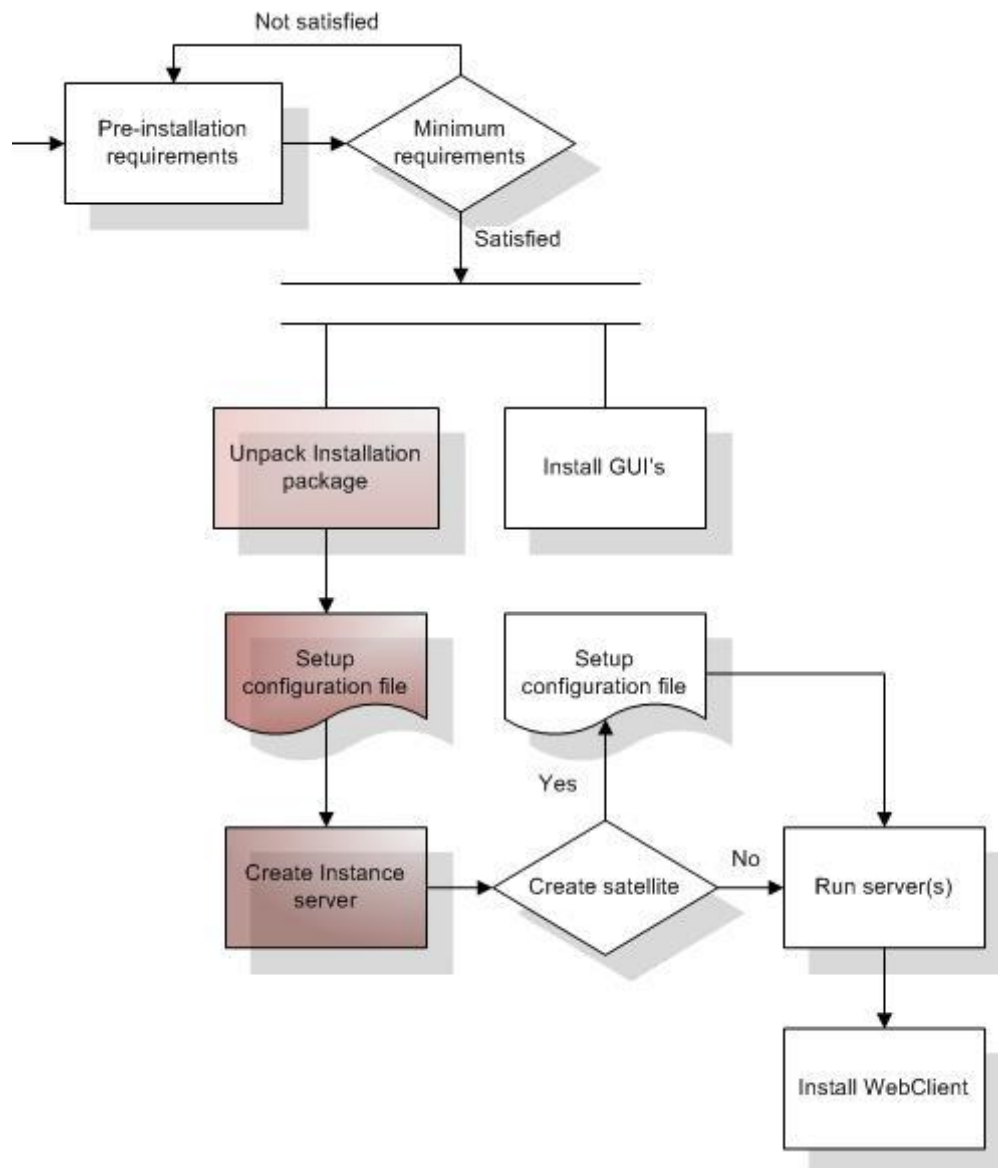


Figure 5 – Server Installation phases

4.1 Software Deployment

Copy the package to a base directory and run the following command:

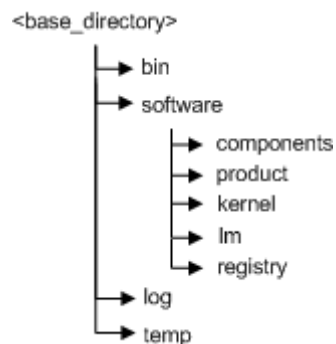
```
> java -jar fraud_8.0.?.?.jar -o <base_directory> -i
```

The following table provides additional options for the command:

Command Parameters	
-h, --help	Show help
-i, --install	The install process
-u, --upgrade	The upgrade process
-l, --language	Default language for instances (defaults to the system language)
-o, --output-dir	The install directory
-r, --region	Instance Region for instances (defaults to system region language)
-v, --verbose	Whether or not to verbose all logging data into output

Table 2 - Additional options for software deployment command

By executing the previous command, a tree is created containing the following directories:



If you want to use the upgrade option, please refer to section 4.4

4.2 Setup Instance Configuration File

When setting up the RAID:FMS server installation, it is possible to create several server instances. These server instances can be from the same or different RAID:FMS modules.

A tool for generating a configuration template file is provided and is located in the **bin** folder. This tool generates a configuration file based on the RAID:FMS predefined configuration. This predefined configuration contains default values for some arguments, therefore allowing the user to use the default values or redefine them.

As mentioned before, each RAID:FMS distribution can contain one or more RAID:FMS modules; therefore, before setting up the configuration file, you must list the available server instance types.

To list the available instance types and licensed modules, execute the following command:

In this example, *TEMP_CUST_all.lic* corresponds to your license file

```
> config-generator -s -l TEMP_CUST_ALL.lic

Copyright (c) WeDo Consulting - Assuring your business for the future
All Rights Reserved

Available Instance Types:
    -> "SatelliteServer" - [Licenced]
        [FMS_Core]
        [FMS_DI]
        [FMS_TS]
        [FMS_ED]
    -> "FraudCenter" - [Licenced]
        [FMS_Core]
        [FMS_DI]
        [ESL_HealthMonitor]
        [FMS_TS]
        [ESL_Admin]
        [FMS_ED] Available Instance Types:
```

The result shows that two instance types are available – **SatelliteServer** and **FraudCenter**.

The following table provides an explanation of the `config-generator` command options:

Command Name	config-generator	
Command Location	<base_directory>/bin	
Command Parameters		Optional
-a,--addon-name	Addon Name	✓
-e,--encoding	Encoding	✓
-h,--help	Help	✓
-s,--show	List instance types <small>Note: This parameter is optional only when you choose instance-type option</small>	✓
-l,--license-file	Used to test license characteristics	✓
-o,--output-file	Output File	✓
-t,--instance-type	Instance Type <small>Note: This parameter is optional only when you choose list option</small>	✓
-v,--verbose	Whether or not to verbose all logging data into output	✓

Table 3 - config-generator command options

To generate the configuration file for a particular instance type, execute the following command:

```
> ./bin/config-generator -t <INSTANCE_TYPE> -o <config_file>.xml
```

The generated configuration file must be edited and the mandatory parameters filled. In addition, the optional (commented) parameters can also be uncommented and edited.

The parameter with the license file is not mandatory. If this parameter is used, then in the generated configuration file there will be a parameter called “feature-groups” that will contain several parameters, with the name of all the feature groups available in your license.

Example:

```
<feature-groups>
  <feature-group name="FMS_Core"/>
  <feature-group name="FMS_ED"/>
  <feature-group name="FMS_DI"/>
  <feature-group name="FMS_TS"/>
</feature-groups>
```

There are 4 feature groups present in the RAID:FMS server installation:

- FMS_Core – FMS Core;
- FMS_ED – FMS Earlier Detection;
- FMS_DI – FMS Deeper Investigation;
- FMS_TS – FMS Tune System.

Like said before, the generated configuration file has a parameter called “feature-groups” that contains feature groups available in your license. If a feature group is removed from the configuration file then it will not be installed (with the exception of FMS_Core, which is ALWAYS installed). If the configuration file used to create the instance has no feature groups, then all feature groups included in the license will be installed.

Some examples, using a license that contains all feature groups, the configuration generator tool creates a configuration file with all 4 feature groups:

Scenario #1: remove feature groups FMS_TS (tune system) and FMS_DI (deeper investigation) from the configuration file. Result: the installation will install Core features and Earlier Detection features.

Scenario #2: remove feature groups FMS_Core, FMS_TS (tune system) and FMS_DI (deeper investigation) from the configuration file. Result: the installation will install Core features and Earlier Detection features (Core ALWAYS installs).

Scenario #3: remove all feature groups from the configuration file. Result: the installation will install all features (if none feature group is present install all included in the license).

The following chapters describe each available parameter – some parameters are only available for specific instance types.

4.2.1 General Parameters

ServerAdminUser

Description	RAID FRAUD administration username.
Data Type	String
Default	adm

ServerHost

Description	RAID FRAUD server name or IP address.
Data Type	String
Default	The value returned by the <code>hostname</code> command.

ServerPort

Description	Unique TCP/IP port for RAID FRAUD application communication.
Data Type	Integer
Range	[1024, 65536]

MailServer

Description	Mail server hostname or IP address. This address must be accessible by the RAID server.
Data Type	String
Default	<i>Empty string</i>

MailServerPort

Description	TCP/IP port for the mail server
Data Type	Integer
Range	[1024, 65536] (must correspond to the port of the mail server to be used)
Default	25

MailServerUser

Description	Authentication user for the mail server.
Data Type	String
Default	<i>Empty string</i>

MailServerPassword

Description	Authentication user's password for the mail server.
Data Type	String Encrypted
Default	<i>Empty string</i>

MailServerUseSSL

Description	Parameter indicating if the communication with the server is made using SSL.
Data Type	Boolean
Range	True or false (case sensitive).

MailServerStartTLS

Description	Parameter indicating if the communication with the server is made using TLS.
Data Type	Boolean
Default	False
Range	True or false (case sensitive).

KeyStorePath

Description	Pathname to the keystore file.
Data Type	String

KeyStorePwd

Description	Password for keystore file.
Data Type	String Encrypted

KeyStoreAlias

Description	All keystore entries are accessed via unique aliases and the KeyStoreAlias will select an entry from keystore to be used in server. If more than one key is present in the keystore it is strongly recommended that a KeyStoreAlias is configured to ensure that the correct key is used.
Data Type	String
Default	afserver

EnableRMFeatures

Description	Indicates if the installation process should enable, or not, the Report Module (RM) component in the installation.
Data Type	Boolean
Range	true or false (case sensitive)
Default	false

EnableRMAdmin

Description	Indicates if the installation process should create, or not, a Web context for Report Module administration web application.
Data Type	Boolean
Range	true or false (case sensitive)
Default	false

xDR_SearchLimit

Description	Maximum number of records a user is able to select by default per datasource type. This is used in the adhoc reconciler component
Data Type	Integer
Default	2000000

UsePartitionedResults

Description	Whether to partition SCH_T_RESULT table which is the storage for all xDR search results
-------------	---

Data Type	Boolean
Default	True

FPUsePartitions

Description	Whether to partition Fingerprinting data
Data Type	Boolean
Default	True

AFDUsePartitions

Description	Whether to partition Advanced Fraud Detection data
Data Type	Boolean
Default	True

AreaPublicName

Description	Public name for this server in the overall server federation
Data Type	String

InternalServerCommunicationUser

Description	User to be used when connecting between servers (e.g. Loading flows)
Data Type	String

InternalServerCommunicationPwd

Description	Password for the internal communication user
Data Type	String Encrypted

ConfigurationAreaDisplayName

Description	Name of the portlet being shown on the portal for the configuration area
Data Type	String

SmartContentDisplayName

Description	Name of the portlet being shown on the portal for the smart content
Data Type	String

LegacyPortletDisplayName

Description	Name of the portlet being shown on the portal for the legacy portlet
Data Type	String
Default	RAID:FMS Legacy (Instance Name)

PortletsCategory

Description	Root node for these portlets in the portlet tree
Data Type	String
Default	RAID:FMS (Instance Name)

CompanyWebIdentifier

Description	Name for the company identification. This must match the one set in the portal
Data Type	String

ExecuteOnExternalProcess

Description	Whether to execute shell scripts on an external process
Data Type	Boolean

ExternalProcessServerPort

Description	Port for the standalone shell script executor service
Data Type	String

GoogleMapKey

Description	Key provided by google sales representative to allow the use of google maps widget
Data Type	String

EnableParallelizationForStateSave

Description	Indicates if parallel jobs are created for the engines' state saving operation.
Data Type	Boolean
Range	true or false (case sensitive)
Default	True

StateSaveMaxThreads

Description	Indicates the number of parallel jobs that are created for the engines' state saving operation. Please be aware that this also implies having a database connection per job.
Data Type	Integer
Default	5

KpiRevenueLossFormula

Description	Indicates the KPI Revenue Loss Formula that should be used.
Data Type	String
Default	0

KpiActiveFrom

Description	Indicates the start day of the KPI Revenue Loss formula.
Data Type	String
Default	01/01/1900

KpiActiveTo

Description	Indicates the last day of the KPI Revenue Loss formula.
Data Type	String
Default	31/12/2900

UpdateReports

Description	Indicates the reports are upgraded when executing an update.
Data Type	Boolean
Default	false

Hourglass_KeepDays

Description	Number of days to keep on the Hourglass service table
Data Type	Integer
Default	10

Hourglass_MaxElements

Description	Maximum number of elements in the run pool
Data Type	Integer
Default	1000

RuleOutputEventLoader_Mode

Description	Event loader for alerts
Data Type	String
Allowed	NONE (no loader), FLOW (use a search flow to load the events' details)
Default	NONE

SubscriptionFraud_TraceMatchToFile

Description	Traces matching details to files during the Subscription Fraud process
Data Type	Boolean
Range	true or false (case sensitive)
Default	false

RuleOptimizerCompressIndicators

Description	Compress rule optimizer auxiliary files (saves space)
Data Type	Boolean
Range	true or false (case sensitive)
Default	True

RuleOptimizerMaxTests

Description	Maximum number of simultaneous Rule Optimizer tests
Data Type	Integer
Default	5

ERTablePrefix

Description	Prefix to be used on Entity Registry tables (may be overridden on application for each entity)
Data Type	String
Default	ER_

SFTablePrefix

Description	Prefix to be used on Subscription Fraud models' tables (cannot be overridden on application)
Data Type	String
Default	SF_

OotbACMBaselineData

Description	Defines if ACM Out Of The Box baseline data is installed
Data Type	Boolean
Range	true or false (case sensitive)
Default	true

OotbDataPumpDataModel

Description	Defines if Data Pump Out Of The Box data model configuration is installed.
Data Type	Boolean
Range	true or false (case sensitive)
Default	false

OotbDataPumpOperationalDashboards

Description	Defines if Data Pump Out Of The Box dashboards are installed.
Data Type	Boolean
Range	true or false (case sensitive)
Default	false

OotbCBPMDDataModel

Description	Defines if CBPM Out Of The Box data model configuration is installed.
Data Type	Boolean
Range	true or false (case sensitive)
Default	false

OotbCBPMOperationalDashboards

Description	Defines if CBPM Out Of The Box dashboards are installed.
Data Type	Boolean
Range	true or false (case sensitive)
Default	false

4.2.2 Database Parameters

DbHost

Description	Oracle database server name or IP address.
-------------	--

Data Type	String
-----------	--------

DbPort

Description	Oracle Database server TCP/IP port.
Data Type	Integer
Range	[0, 65536]

DbServiceName

Description	Oracle Database Service Name.
Data Type	String

DbSid

Description	Oracle Database System Identifier.
Data Type	String
Default	<DbServiceName>

DbJdbcUrl

Description	Oracle database JDBC URL. When filled redefines the default URL.
Data Type	String
Default	jdbc:oracle:thin:@<DbHost>:<DbPort>:<DbSid>

DbUsersDbUser

Description	DBA user name. Used for creating application database users.
Data Type	String
Default	Empty

DbUsersDbUserPw

Description	Database administration user password.
Data Type	String
Default	String Encrypted

DbUsersAdmUser

Description	Database administration user name that owns RAID Core objects.
Data Type	String

DbUsersAdmPw

Description	Database administration user password.
Data Type	String Encrypted

DbUsersDatUser

Description	Database user that will own all non RAID:FMS objects. If the user is the same as the ADM user (default), all objects are created in the ADM user.
Data Type	String
Default	<DbUsersAdmUser>

DbUsersDatPwd

Description	Database DAT user password.
Data Type	String Encrypted
Default	<DbUsersAdmPwd>

DbUsersAppUser

Description	Database application user name that accesses all objects (no schema manipulation access).
Data Type	String

DbUsersAppPwd

Description	Database application user password.
Data Type	String Encrypted

DbTablespacesSmallData

Description	Tablespace for small tables.
Data Type	String

DbTablespacesMediumData

Description	Tablespace for medium tables.
Data Type	String

DbTablespacesLargeData

Description	Tablespace for large tables.
Data Type	String

DbTablespacesSmallIndex

Description	Tablespace for small indexes.
Data Type	String

DbTablespacesMediumIndex

Description	Tablespace for medium indexes.
Data Type	String

DbTablespacesLargeIndex

Description	Tablespace for large indexes.
Data Type	String

DbTablespacesLOB

Description	Tablespace for LOB objects.
Data Type	String

DbTablespacesTemp

Description	Temporary tablespace.
-------------	-----------------------

Data Type	String
Default	TEMP

DbTablespacesDefault

Description	Default user's tablespace.
Data Type	String

DbTablespacesQuota

Description	A list of additional tablespaces that you can give unlimited quota.
Data Type	Array

DbCreateDATUser

Description	Whether or not to create DAT user.
Data Type	Boolean
Default	If <code>Adm=Dat user</code> , defaults to false.
Range	true or false (case sensitive).

DbCreateUsers

Description	Whether or not to create users.
Data Type	Boolean
Default	If <code>Db a</code> variable not empty, defaults to true.
Range	true or false (case sensitive).

StatisticsDataTablespace

Description	Tablespace for statistics tables.
Data Type	String
Default	If not defined, defaults to <i>DbTablespacesLargeData</i>

StatisticsIndexTablespace

Description	Tablespace for statistics indexes
Data Type	String
Default	Defaults to <i>DbTablespacesLargeIndex</i>

AlertsDataTablespace

Description	Tablespace for alerts tables.
Data Type	String
Default	Defaults to <i>DbTablespacesLargeData</i>

AlertsIndexTablespace

Description	Tablespace for alerts indexes
Data Type	String
Default	Defaults to <i>DbTablespacesLargeIndex</i>

CasesDataTablespace

Description	Tablespace for cases tables.
Data Type	String
Default	Defaults to <i>DbTablespacesLargeData</i>

CasesIndexTablespace

Description	Tablespace for cases indexes
Data Type	String
Default	Defaults to <i>DbTablespacesLargeIndex</i>

4.2.2.1 Scheduler component

The scheduler configuration can be shared across different product installations.

Usually, each instance will have its own scheduler configurations. If that is the case, the parameters mentioned below can be left with the defaults, which will maintain the Scheduler configurations along with the rest of the product settings.

If, however, you wish to share the Scheduler settings with an already installed product, please redefine these parameters to point to an already installed database, where those settings are maintained.

Scheduler DB parameter	Default
SchedulerDbHost	DbHost
SchedulerDbPort	DbPort
SchedulerDbServiceName	DbServiceName
SchedulerDbSid	SchdulerDbServiceName
SchedulerDbJdbcUrl	jdbc:oracle:thin:@<SchedulerDbHost>:<SchedulerDbPort>:<SchedulerDbSid>
SchedulerDbUsersAdmUser	DbUsersAdmUser
SchedulerDbUsersAdmPwd	DbUsersAdmPwd
SchedulerDbAppUser	DbUsersAppUser
SchedulerDbAppPwd	DbUsersAppPwd

Note: Each Scheduler DB Parameter has the same characteristics as the Defaults.

For example: SchedulerDbHost is equivalent to DbHost:

Description	Oracle database server name or IP address.
Data Type	String

Check 4.2.2 - Database Parameters for further details.

4.2.3 Server Parameters

ServerMemoryOptions

Description	Initial and Maximum memory allocation for RAID FMS server (Java process).
Data Type	String

Default	-Xms128M
(single line)	-Xmx1024M
	-XX:+UseParNewGC
	-XX:+UseConcMarkSweepGC
	-XX:+CMSParallelRemarkEnabled
	-XX:-DisableExplicitGC
	-XX:MaxPermSize=256m

ConnMngrMaxConnections

Description	Maximum number of simultaneous open connections to RAID database.
Data Type	Integer
Default	50
Range	1 – maximum number of open connections allowed by the database.

ConnMngrVerbose

Description	Indicates if the queries being executed in the database should be outputted.
Data Type	Boolean
Default	false
Range	true or false (case sensitive).

ConnMngrDatMaxConnections

Description	Maximum number of simultaneous open connections to RAID DAT database.
Data Type	Integer
Default	50
Range	1 – maximum number of open connections allowed by the database.

ConnMngrAdmMaxConnections

Description	Maximum number of simultaneous open connections to RAID ADM database.
Data Type	Integer
Default	5
Range	1 – maximum number of open connections allowed by the database.

DataModelMaxSharedConnectionsPerUser

Description	Maximum number of shared connections per user, for Data Model
Data Type	Integer
Default	5
Range	1 – 10

MaxSessions

Description	Maximum number of sessions allowed in the RAID-FRAUD server.
Data Type	Integer
Default	30

SessionTimeout

Description	Time, in milliseconds, after which an idle session closes.
Data Type	Long
Default	1800000

ServerUseSSL

Description	Indicates if the RAID server should use Secure Sockets Layer (SSL).
Data Type	Boolean
Default	false
Range	true or false (case sensitive).

MasterInstanceId

Description	The name of the master instance in case this is a satellite instance.
Data Type	String
Default	Defaults to current \$InstanceId

EnableDBJavaUtils

Description	Enable the installation of some Java utilities for database use.
Data Type	Boolean
Default	False

4.2.4 Auditing Parameters

AuditLogLineFormat

Description	Define the order and separation between fields by which they are presented in audit file. You only need to set this parameter if you want to override the default value, otherwise do not specify the parameter in the configuration file.
Data Type	String
Default	{0},{1},{2},{3},{4},{5},{6},{7},{8},{9},{10},{11},{12},{13},{14},{15},{16},{17}

4.2.5 Portal Integration

Note: The context names cannot contain space characters

WebServerHost

Description	Tomcat web server hostname or IP address. This address must be accessible and known (DNS-wise) by all machines that use the web client application.
Data Type	String
Default	The value returned by the <code>hostname</code> command.

WebServerPort

Description	Tomcat web server port.
Data Type	Integer
Range	[0, 65536]

FraudCenterContextName

Description	Tomcat web context name hosting the RAID FRAUD legacy (AB) web client
Data Type	String
Default	<code>Rmadmin</code>

RMAdminContextName

Description	Tomcat web context name hosting the RAID Report Module Administrator web client.
Data Type	String
Default	<code>Rmadmin</code>

PathToPortalInstance

Description	The path for the portal instance where to deploy product artifacts (e.g. <code>/instances/<PRT instance></code>). If not specified a manual step is required to deploy the web artifacts in the portal
Data Type	String
Default	

WebServerUseSSL

Description	Indicates if the RAID web server should use SSL.
Data Type	Boolean
Default	<code>false</code>
Range	true or false (case sensitive).

BackendServerHost

Description	Backend hostname to use on internal connections
Data Type	String

Default	<current host>
---------	----------------

PortalDbServiceName

Description	Portal database service name.
Data Type	String
Default	If not specified the default is <code>\${PortalDbServiceName}</code> .

PortalDbJdbcUrl

Description	Portal JDBC URL
Data Type	String
Default	<code>jdbc:oracle:thin:@\${PortalDbHost}:\${PortalDbPort}:\${PortalDbSid}</code>

PortalDbHost

Description	Database hostname. Mandatory if JDBCUrl is not used.
Data Type	String
Default	Not needed if <code>\${DbJdbcUrl}</code> is supplied.

PortalDbPort

Description	Database port. Mandatory if JDBCUrl is not used.
Data Type	String
Default	

PortalDbSid

Description	The database SID. Mandatory if JDBCUrl is not used.
Data Type	String
Default	Defaults to <code>\${DbServiceName}</code> if no value is supplied.

PortalDbUsersAppUser

Description	Database application user to backend database.
Data Type	String
Default	

PortalDbUsersAppPwd

Description	Database password to backend database.
Data Type	String Encrypted
Default	

PortalDbUsersAdmUser

Description	Database ADM user to backend database.
Data Type	String
Default	

PortalDbUsersAdmPwd

Description	Database ADM password to backend database.
Data Type	String Encrypted
Default	

Note: In an installation where the database is using the RAC architecture, this parameter (PortalDbJdbcUrl) must **always** be redefined. If it is the same as the DbJdbcUrl, a reference may be used (<ref name="PortalDbJdbcUrl" refer-to="DbJdbcUrl" />), otherwise the full configuration must be filled.

4.2.6 Change Server Parameters after Installation

After installing a RAID:FMS server instance, situations may occur where it is necessary to change the server parameters. In order to do this, the RAID:FMS installation process provides a script that allows refreshing changed configuration parameters.

The explanation of this script, available options and a simple example are described below.

Command Name	apply-instance-config	
Command Location	<base_directory>/bin	
Command Parameters		Optional
-b, --bypass-unit-list	List of units to ignore during the update (e.g. af.DbModel, af.Access)	✓
-c, --config-file	Changed configuration file to be used Note: This parameter is optional only when you have previously changed other configuration	✓
-f, --force-unit-list	List of units to force during the update (e.g. af.DbModel, bpm.DbModel). This is a unit bypass but registering the unit as completed	✓
-h, --help	Help	✓
-n, --instance-name	Instance where the new configuration parameters will be applied	
-v, --verbose	Whether or not to verbose all logging data into output	✓

Table 4 - apply-instance-config command options

To update a server instance configuration, execute the following command:

```
> ./bin/apply-instance-config -c <ChangedConfigurationFile> -n <InstanceName>
```

The license file parameter is optional in this command. It should be used if the license file has changed since the installation.

If the new license has new features that were not present the previous license then these new features will be installed.

If the new license has less features than the previous license then the deprecated features will be inactivated and will not be available in the server after executing this command.

4.2.7 Protecting passwords

The `encrypt-text` script is used to obscure the text and ensure privacy. The available parameters for this script are described in the following table:

Name	<code>encrypt-text</code>	
Location	<base_directory>/bin	
Options	Description	Optional
<code>-h, --help</code>	Prints the help	✓
<code>-s, --silent <Boolean></code>	Silent mode. Only prints the encrypted text.	✓
<code>-t, --text <String></code>	The text to be encrypted	

Table 5 - `encrypt-text` Command Parameters

Below is an output example for the execution of this operation.

```
> ./encrypt-text -t Password1
Text encryptor
Copyright (c) WeDo Consulting - Assuring business for the future
All Rights Reserved

9ADE26D7D93535B95E71B107A4D8B4FCB620BD441B4B7EF51E91F2DAD02360A4
```

The *encrypted* type argument can easily be defined, using *ConfigMap*, with the following XML code:

```
<encrypted name="ArgumentName">EncryptedValue</encrypted>
```

Note: When you need to use special characters in the text to encrypt, you can define the text argument for the tool between `"`. The tool only encrypts the text, ignoring the `"` (in double quotes). But in some cases, the characters: `"`, `$`, ```, `!` and `\` are still interpreted by the shell, even when they're in double quotes. The backslash (`\`) character is used to mark these special characters so that they are not interpreted by the shell, but passed on to the `encrypt` command.

4.3 Create and Run RAID:FMS Instance Server

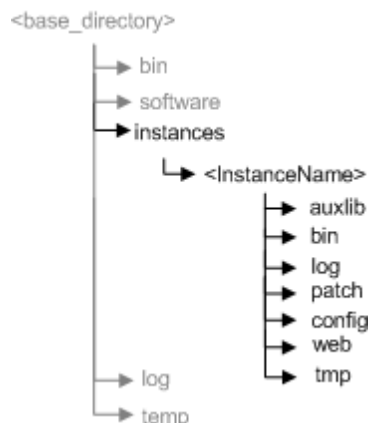
The next section describes how to create and run a RAID:FMS instance server.

Before creating an instance server, complete all the steps listed in section 2.2.

To create a RAID:FMS instance server, execute the *create-instance* script in the `bin` directory indicating the configuration file generated in section 4.2, as argument:

```
> ./bin/create-instance -c <config_file> -n <InstanceName> -t <InstanceType> -l
<licence file>
```


As a result, the following tree is created:



To generate the *DDL* file (that contains the statements that create database users, roles and necessary objects) used in the [Advanced](#) Installation mode, execute the following command:

```
> ./bin/create-instance -c <config_file> -d <DDL_Directory> -n <InstanceName> -t
<InstanceType> -l raidfms.lic
```

Note: The *DDL_Directory* must be a regular directory with writing permission

License files should be obtained from the License Support Desk.

4.3.1 Run RAID:FMS Instance Server

To start the server, change to the recently created *<InstanceName>* directory and execute:

```
> ./bin/start
> ./bin/start-search
```

Check that all components have been initialized in the log file created under the logs directory.

Note: The start server log file looks like this:

<InstanceName>_server_YYYYmmDD_hhmmss.log


The following table provides an explanation of the *create-instance* command options:

Command Name	create-instance	
Command Location	<base_directory>/bin	
Command Parameters		Optional
-b, --bypass-unit-list	List of units to bypass during the update (e.g. af.DbModel, bpm.DbModel)	✓
-c, --config-file	Configuration File to be used	
-l, --licence-file	The licence file to supply	
-d, --log-ddl	Log DDL statements to a log file	✓

<code>-f, --force-unit-list</code>	List of units to force during the update (e.g. af.DbModel, bpm.DbModel). This is a unit bypass but registering the unit as completed	✓
<code>-g, --upgrade-unit-list</code>	List of units to force upgrade (e.g. af.DbModel, af.Access)	✓
<code>-h, --help</code>	Help	✓
<code>-i, --ignore-ddl</code>	Ignore DDL statements	✓
<code>-l, --instance-language</code>	Language (e.g. en, pt)	✓
<code>-n, --instance-name</code>	Instance Name	
<code>-r, --instance-region</code>	Instance Region for instances (defaults to system region language)	✓
<code>-t, --instance-type</code>	Instance Type	
<code>-v, --verbose</code>	Whether or not to verbose all logging data into output	✓

Table 6 - create-instance Command options

4.4 Upgrading the RAID:FMS Server

This chapter is dedicated to the upgrade of the RAID:FMS server, but this only applies to patch or build level upgrades (in the RAID:FMS version we have “major.minor.patch.build”). If your current version is 8.0, you can only upgrade to newer versions of 8.0.x.x. For upgrades between versions with different major or minor (from version 5.1 to a 7.1), please refer to the Configurations Migration Guide  [CMG80].

This upgrade is performed in two steps; first, upgrade your current software version with the jar file of the most recent version, and second, update all installed instances with the new changes.

To perform the first step (software update), please execute the following UNIX command:

```
> java -jar raid_fraud-8.0.?.?.jar -o <base_directory> -u
```

After issuing this command, a backup of all software is created in the `update` directory in the `<base_directory>` and all changed components are replaced. After this step, all instances are marked as dirty and you should refresh them with the `update-instance` command. For more details about this command please refer to section 4.1.

The second step (instance update) needs to be performed once for each instance, as all instances need to be updated in order to function properly. Execute the following command for each instance you have installed:

```
> ./bin/update-instance -n <InstanceName> -l <Licence File>
```

The following table provides an explanation of the `update-instance` command options:

Command Name	update-instance	
Command Location	<base_directory>/bin	
Command Parameters		Optional
-b, --bypass-unit-list	List of units to bypass during the update (e.g. af.DbModel, bpm.DbModel)	✓
-c, --config-file	Configuration File to be used. This is only needed if the new version requires more config parameters than the previous. This should be a copy of product-config.xml with the extra parameters.	✓
-l, --license-file	The new license file to be used	✓
-d, --log-ddl	Log DDL statements to a log file	✓
-f, --force-unit-list	List of units to force during the update (e.g. af.DbModel, bpm.DbModel). This is a unit bypass but registering the unit as completed	✓
-g, --upgrade-unit-list	List of units to force upgrade (e.g. af.DbModel, af.Access)	✓
-h, --help	Help	✓
-i, --ignore-ddl	Ignore DDL statements	✓
-l, --instance-language	Language (e.g. en, pt)	✓
-n, --instance-name	Instance Name	
-r, --instance-region	Instance Region for instances (defaults to system region language)	✓
-v, --verbose	Whether or not to verbose all logging data into output	✓

Table 7 - update-instance Command options

4.5 Uninstall Server

4.5.1 Uninstall Process

RAID:FMS provides an uninstall script that allows you to remove a RAID:FMS server instance. The script can be found in <base_directory>/bin.

A brief description of the script available parameters and execution example can be found below:

Command Name	drop-instance	
Command Location	<base_directory>/bin	
Command Parameters		Optional
-d, --log-ddl	Log DDL statements to a log file	✓
-h, --help	Help	✓
-i, --ignore-ddl	Ignore DDL statements	✓
-n, --instance-name	Instance Name	
-v, --verbose	Whether or not to verbose all logging data into output	✓

Table 8 - drop-instance Command Parameters and execution example

In order to remove a RAID:FMS server instance, execute the following command:

```
> ./bin/drop-instance --instance-name=<InstanceName>
```

The drop-instance script is responsible for removing the specified instance and related database objects.

Note: *It is strongly advised that you ensure any files or directories under <InstanceName> that are not in use.*

4.5.2 Installation Error

If a problem occurs during server installation, the RAID:FMS instance database schema objects and users must be removed. To do that, the uninstall script should be used to remove database schema objects and users.

```
> ./bin/drop-instance --instance-name=<InstanceName>
```

4.5.3 Reusing DAT Schema in Upgrade

Note: This section only applies to installations where the DAT schema was created separately from the ADM schema.

Upgrading RAID software requires redeploying database objects from the new version. Two approaches can be used to upgrade RAID:

- Reuse database schemas: this approach implies uninstalling database schemas without deleting DAT schema and using the new RAID version installation process to redeploy them, as well as to configure the new RAID to reuse the DAT schema;
 In order to uninstall database schemas without deleting DAT schema, the uninstall script should be used with the `--log-ddl` option. This option writes the database objects removal statements into a file. The removal statements should then be selected, in order to remove all database objects, except the ones concerning the DAT schema.
- Use different database schemas: this approach only requires configuring the new RAID version to use the existing DAT schema.

Note: Use the import/export tool bundle with RAID server to migrate RAID data from the old installation into the new one. For further information, please consult the manual [\[CMG80\]](#)

4.6 System Properties Configuration

Some system properties configuration can be used to change the application behavior. These can be defined by adding to the environment variables the following variable:

```
<InstanceName>_SERVER_JAVA_OPTS=-D<parameter>=<value>
```

More than one parameter can be changed.

4.6.1 General Properties

wedo.jaf.services.xdt.XdtResponder.trace

Description	Logs all sent and received requests.
Data Type	Boolean
Default	false
Range	true or false (case sensitive)

4.7 SSL Configuration

RAID servers can use *SSL* in their communications and this section describes how to do it. Please note that it's advisable to use trusted certificates in this process to completely ensure the identity of all the nodes in the system (especially on the webserver side).

4.7.1 Basics

To effectively configure an *SSL* environment two things (at the very minimum) are needed:

- Key store
- Certificate

The recommended and simplest configuration, and the one we'll be describing here, is the use of a single key store for all servers in the same machine.



To enforce the maximum security it's advisable to use certificates issued by a trusted certification authority. Self-signed certificates may be used but they are not advised.

4.7.1.1 Key store

To generate a key store issue the following command:

```
> keytool -genkey -keyalg rsa -keypass <ks-password> -alias <key-alias> -storepass <ks-password> -keystore <path-to-ks-file> -dname "CN=<server-name>"
```

The **<server-name>** is the exact name of the hostname where the server is installed and in accordance with `ServerHost` parameter, described in 4.2.1 section.

E.g. when trying to access a server named **myserver** then **server-name=myserver**.


4.7.1.2 Certificate

To export the public key (*.crt*) for the generated key store issue the following command:

```
> keytool -export -alias <key-alias> -keystore <path-to-ks-file> -rfc -file <path-to-crt-file>
```

After the export step, it's advisable to add the certificate in all the JREs trust store. The default cacerts password is "changeit".

```
> /keytool -import -alias <key-alias> -file <path-to-crt-file> -keystore
$JAVA_HOME/lib/security/cacerts
```

	<p>The import alias, <key-alias>, must be new in the cacerts keystore.</p> <p>Because the generated certificated is not trusted, it's advisable to add it to all trust stores for all the servers where the product is installed</p>
---	--

4.7.2 Application server

The application server is expecting a key store which contains a specific certificate with alias. The alias should be defined at certificate creation.

For each server instance specify the following parameters in the configuration file:

- `<boolean name="ServerUseSSL">true</boolean>` This tells the *Application* server that it should use SSL in all communications;
- `<string name="KeyStorePath"><path-to-ks-file></string>`
 - This is the path to the key store to be used;
 - It should be an absolute path accessible by all servers ;
- `<string name="KeyStorePwd"><ks-password></string>`
 - The password of the previously specified key store;
- `<string name="KeyStoreAlias"><key-alias></string>`
 - This parameter tells the *Application* server which certificate should be used from keystore. All keystore entries are accessed via unique aliases;
- For the *Application* server to be able to communicate with other *Application* servers you need to add the generated certificate to the Java trusted key store.
 - This is to be performed by the system administrator;
 - If for some reason (e.g. simple tests) you cannot add the certificate directly to the *Java* trusted key store you can set the following parameter in your configuration file to override the default *Java* trusted key store by your own:
 - `<string name="ServerMemoryOptions">-Xms128M -Xmx1024M -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled -XX:MaxPermSize=256m -Djavax.net.ssl.trustStore=<path-to-ks-file> -Djavax.net.ssl.trustStorePassword=<ks-password></string>`
 - Even though this parameter is to be used to specify server memory options you can also specify other *JVM* arguments in here;
 - To be able to execute command line scripts (e.g. *bcm-expand-entity*) with `-ssl true` you should also configure the trusted store in the `<product-install>/software/kernel/config/software.conf.ext` file. If the file doesn't exist simply create it. Add the following line to the end of this file:


```
SCRIPT_JAVA_OPTS="${SCRIPT_JAVA_OPTS} -Djavax.net.ssl.trustStore=<path-to-ks-file> -Djavax.net.ssl.trustStorePassword=<ks-password>"
```

4.7.3 Application Browser

Obtain the previously generated certificate (.crt) and **install it** in the **trusted certificates store** of the *MS Windows* system where the *Application Browser* is installed.

Failing to do the above step successfully prevents *Application Browser* to access *Application* servers that are using SSL.

Finally, configure the `appbrowser.properties` file with *HTTPS URLs*.

```
<description> = https://<server-name>:<port>/<uri>
```

Remember that `<server-name>` must be exactly what was specified when generating the key store.

5 Auditing Features

5.1 Auditing Log

RAID:FMS supports the Auditing Log. To enable the use of this feature, it is necessary to define at least one audit filter. RAID:FMS provides one audit filter file for all relevant administrative operations present in `/config/audit` directory.

Note: Location that contains all Filter Files is a relative path relative to the instance server directory.

This location contains all *filter files* to be used by the auditing service.

The auditing service provides a mechanism to define or redefine auditing filters. Each *filter file* present in the filters location defines a set of filters and its precedence. Filter precedence defines an order by which they are loaded. One filter with precedence $n+1$ can redefine a filter with precedence n .

5.1.1 Audit Logger File

The Audit Manager is responsible for gathering all events and writing them into a log file defined in **AuditOutputFile**. Each line of these files can be configured. In **AuditLogLineFormat** (explained in section 4.2.4), each “{d}” represents a field, according to the following table.

Field Name	Field Number	Description
Client Type	{0}	A free text filled by the client when creating a connection. This field should contain what kind of client is connecting, like Application Browser (AB), Web Client (AFWC), and so forth.
Client Version	{1}	The client version used to establish the connection.
Client Username	{2}	The username logged in the operating system where the client is running.
Remote IP	{3}	The IP address of the connection received by the server.
Remote Hostname	{4}	The hostname of the connection received by server or the IP address if the hostname cannot be determined.
Client IP	{5}	The IP address given by the client.

Client Hostname	{6}	The hostname given by the client or the IP Address if the hostname is unable to be determined. If reverse DNS lookup fails for a given IP address this field will always show the IP address. Note: The use of this field can affect server response performance due to the need of extra DNS lookups. ¹
Referer	{7}	The web address (URL) where the web client triggered the event (applicable only to web clients).
Application Name	{8}	The name of the application where the event occurred.
Username	{9}	The AF username that triggered the activity.
SID	{10}	The AF session identifier.
Activity Type	{11}	The activity type of the event.
Activity Name	{12}	The business activity name being tracked (operation name).
Activity Description	{13}	Business activity information. This field shows the activity parameters (for example, on a flow modification this can show the flow identifier).
Result	{14}	The outcome result of the event execution.
Error Code	{15}	The error code associated with the Failure Description field.
Failure Description	{16}	Error message given on failure events.
Operation Time	{17}	The execution time of the operation in milliseconds

Table 9 - Audit Log Line fields

Regardless of the log line configuration, all logger lines start with a timestamp value.

To provide the final user with more control over the **Activity Description** message, this can be a template text that is parsed in each event execution. The template language uses **Velocity** [1.6 version] scripting language (supplied by Apache Software Foundation) and has access to the input and output content of the event (operation), except for system events. The input is always accessible using the **Input** variable and the output using the **Output** variable.

¹ In Tomcat set the **enableLookups** connector attribute to **false** to skip the DNS lookup and return the IP address in String form instead (thereby improving performance). By default, DNS lookups are enabled.

Sample:

Operation: *Adm.ModifyUserStatusLockUnlock*

User: *UserSample*

Content Template Message for Activity Description:

```
#if (${Input.ActionLock} == "lock")
User [ ${Input.Username} ] locked. Reason: ${Output.Lockreason}
#else
User [ ${Input.Username} ] unlocked
#end
```

Result in audit file:

```
User [UserSample] locked. Reason: Security reasons
```

Note: For more details about this scripting language syntax, check the site:

<http://velocity.apache.org/engine/devel/user-guide.html>.

Additionally, a utility object is made available in the template parsing, named **Utils**, that aggregates a set of utility methods to aid the user obtaining the desired result. For now, the only method available is **pMap2XML(ParameterMap)**. This method allows converting a ParameterMap object into a XML string representation.

Sample:

Input

```
field1 = "TEST"
```

Template Message

```
${Utils.pMap2XML(${Input})}
```

Result

```
<parameter name="field1" type="String">TEST</parameter>
```

5.1.2 Filter File

To configure a message that describes an activity **Content**, the input is always accessible using **Input** variable and the output using **Output** variable. The following table shows a set of relevant fields for each variable:

Operation Name	Relevant Input Fields	Relevant Output Fields
CreateUser	Username (<i>String</i>) Name (<i>String</i>)	
ChangePassword	Username (<i>String</i>)	
ModifyUser	Username (<i>String</i>) Name (<i>String</i>) NeverExpires (<i>Boolean</i>) ForcePasswordChange (<i>Boolean</i>)	

DeactivateUser	Username (<i>String</i>)	
AddUserProfile	Username (<i>String</i>) ProfileId (<i>String</i>) ApplicationId (<i>String</i>)	
RemoveUserProfile	Username (<i>String</i>) ProfileId (<i>String</i>) ApplicationId (<i>String</i>)	
SetProfilePermissions	ProfileId (<i>String</i>) PermissionList (<i>Array</i>) ApplicationId (<i>String</i>)	
ModifyPassword	Username (<i>String</i>)	
DeleteUser	Username (<i>String</i>)	
ModifyUserStatusLockUnlock	Username (<i>String</i>) ActionLock (<i>String</i>)	Lockreason (<i>String</i>)
SetUserProfiles	Username (<i>String</i>) ProfileList (<i>Map</i>)	

Table 10 - Operation Relevant fields

The following sections describe the structure of the file filter, and present a simple sample.

5.1.2.1 Base Definition

```
<parameter type="Map">
  <!--
    Filter precedence define an order by which they are loaded.
    One filter with precedence n+1 can redefine a filter with precedence n
  -->
  <parameter name="Precedence" type="Integer">#FilterPrecedence#</parameter>

  #SystemEvents# (0..1)

  <parameter name="OperationFilters" type="Array">

  #OperationFilter# (0..n)

</parameter>
</parameter>
```

5.1.2.2 System Events

```
<parameter name="SystemEvents" type="Map">
  <!--
    For register every Login(s) in the system
  -->
  <parameter name="Login" type="Boolean">true|false</parameter>

  <!--
    For register every Logout(s) in the system
  -->
  <parameter name="Logout" type="Boolean">true|false</parameter>
```

```
<!--
  For register every startup(s) of the system
-->
<parameter name="StartServer" type="Boolean">true|false</parameter>

<!--
  For register every shutdown(s) of the system
-->
<parameter name="StopServer" type="Boolean">true|false</parameter>
</parameter>
```

5.1.2.3 Operation Filter

```
<parameter type="Map">

<!--
  Expression that define a operation (set) that will be apply this filter
-->
<parameter name="Match" type="String">#OperationName#|#OperationSetMatch#</parameter>

<!--
  Literal - must be used when parameter Match definition is a operation name
  Regexp - must be used when parameter Match definition is a regular expression
-->

<parameter name="MatchType" type="String">Literal|Regexp</parameter>

<!--
  StartOperation - log activity before the operation execution
  EndOperation   - log activity after the operation execution
  All            - log activity after and before the operation execution
-->

<parameter name="EventType" type="String">StartOperation|EndOperation|All</parameter>

<!-- The event is or not to registered in audit file -->
<parameter name="Log" type="Boolean">true|false</parameter>

<!-- Customized message to describe a activity -->
<parameter name="Content" type="String">#Velocity Template#</parameter>
</parameter>
```

5.1.2.4 Filter Sample

Simple *Filter* sample, with all parameters:

```
<parameter type="Map">
  <parameter name="Precedence" type="Integer">0</parameter>
  <parameter name="SystemEvents" type="Map">
    <parameter name="Login" type="Boolean">true</parameter>
    <parameter name="Logout" type="Boolean">true</parameter>
    <parameter name="StartServer" type="Boolean">true</parameter>
    <parameter name="StopServer" type="Boolean">true</parameter>
  </parameter>
  <parameter name="OperationFilters" type="Array">
    <parameter type="Map">
      <parameter name="Match" type="String">Adm.SetUserProfiles</parameter>
      <parameter name="MatchType" type="String">Literal</parameter>
      <parameter name="EventType" type="String">EndOperation</parameter>
      <parameter name="Log" type="Boolean">true</parameter>
      <parameter name="Content" type="String">
The user [{Input.Username}] has a new profile list:
  #foreach($profile in $Input.ProfileList.entrySet())
    ${profile.value.ApplicationId}.${profile.value.ProfileId}
```

```

        #end
        </parameter>
    </parameter>
    <parameter type="Map">
        <parameter name="Match" type="String">Adm.ModifyUserStatusLockUnlock</parameter>
        <parameter name="MatchType" type="String">Literal</parameter>
        <parameter name="EventType" type="String">EndOperation</parameter>
        <parameter name="Log" type="Boolean">true</parameter>
        <parameter name="Content" type="String">
            #if (${Input.ActionLock} == "lock")
User [${Input.Username}] locked. Reason: ${Output.Lockreason}
            #else
User [${Input.Username}] unlocked
            #end
        </parameter>
    </parameter>
</parameter>
</parameter>
</parameter>

```

The filter shown above defines filters to audit the operations ***Adm.SetUserProfiles***, ***Adm.ModifyUserStatusLockUnlock***, and all system events.

The ***Content*** message in the first filter is based on the ***ProfileList*** iteration. The second case defines two different messages for ***Content*** according to the ***ActionLock*** value.

6 Installing a RAID:FMS Satellite Server

The RAID:FMS Satellite Server supports the FRAUD main server, offering additional processing power in order to deliver load balancing and improve performance. If required, additional RAID:FMS Satellite servers can be deployed to achieve the desired performance.

This type of server uses the same database schemas of the RAID:FMS main server to read all configurations and therefore is always bound to a specific FRAUD main server.

The steps for creating a new RAID:FMS Satellite Server instance are identical to the ones used in a RAID:FMS main server installation, with a few differences.

6.1 Setting up the Instance Configuration File

Generate a new Satellite Server instance configuration file:

```
> ./bin/config-generator -t SatelliteServer -o <satConfigFile>
```

Edit and review the following parameters:

ServerPort

Description	Unique TCP/IP port for administration and client applications communication.
Data Type	Integer
Range	[1024, 65536]

MasterInstanceId

Description	The instance name of the master instance. The master instance was the first RAID:FMS Server to be installed in the cluster
Data Type	String

PerformServerRegistration

Description	Whether or not to register the satellite in the portal.
Data Type	Boolean
Default	False
Range	true or false (case sensitive).

Note1: All the remaining parameters are the same as the ones used in the master instance (See section 4.2).

Note2: It is recommended that the configuration **PathToPortalInstance** should not be defined or the satellite instance also deploys war files to the portal server.

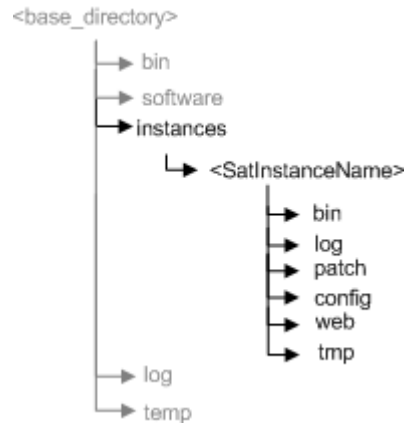
Note3: The use of the license file to create the configuration file for a Satellite Server works the same as for the RAID-FRAUD server, see Chapter 4.2 Setup Instance Configuration File.

6.2 Creating an Instance of a RAID:FMS Satellite Server

To create a RAID:FMS instance satellite, you must execute the `create-instance` script in the `bin` directory indicating the satellite configuration file as the argument:

```
> ./bin/create-instance -c <satConfigFile> -n <SatInstanceName> -t <InstanceType>
-l <LicenceFile>
```

As a result, the following tree is created:



To start the satellite server, change to the recently created `<SatInstanceName>` directory and execute:

```
> ./bin/start
```

Check that all components have been initialized in the log file created under the `logs` directory.

Note: The start server log file looks like this:

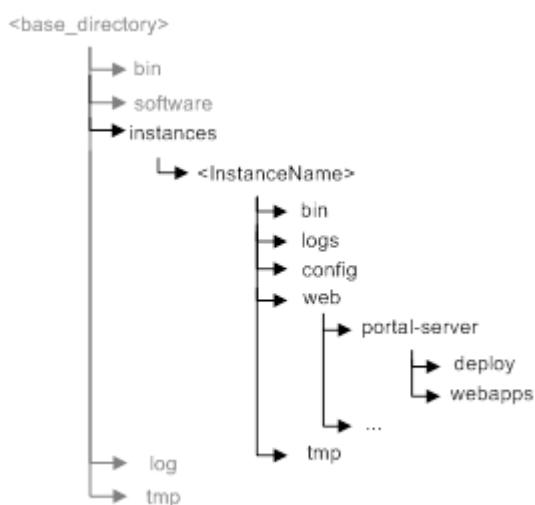
```
<SatInstanceName>_server_YYYYmmDD_hhmmss.log
```

7 Deploying Application Portlets

The product provides a mechanism to auto-deploy all the portlets/web-applications created during the installation. This is the preferred deployment method. To activate this, you only have to supply the **PathToPortalInstance** parameter in your product configuration.

```
<string name="PathToPortalInstance">[portal_deployment]/instances/[instanceName]</string>
```

If you prefer to manually deploy the portlets/webapplications/help in a local or a remote portal deployment, you need to copy all the files and directories contained in the web/portal-deploy directory to the target folder.



To copy web artifacts manually to your Portal, execute the following command:

```
#cp -R [raid_deploy]/instances/[instanceName]/web/portal-deploy/* [portal_deployment]/instances/[instanceName]/portal-server_1.0/
```

Note: If portal deployment is in a remote location, you can copy this file using any available protocol or filesystem (e.g. FTP, SFTP, NFS ...).

8 How to Install an Add-on

This chapter describes the steps required to deploy and install an add-on. Currently, only the add-ons listed in the following table are available.

Add-on Name	Description/Usage
fms_ui (Fraud User Interface)	Adds fraud user interface to CM. This add-on must be installed under the CM instance (Case Management Area).

Table 11 - List of Add-ons

NOTE: The following steps are performed in the CM software location and instance(s).

1. Stop instance server(s)

Before executing any installation step of the add-on, you must make sure that CM instance server is stopped. To stop the instance server, you must execute the following command line in the instance base directory. ***This step is only applicable to products with instances installed.***

```
> ./<CM software location>/instances/<instance name>/bin/stop
```

2. Deploy add-on software

The first step to install the add-on is to deploy its software into the product by executing the next command:

```
> ./<CM software location>/bin/addon-software-install -p <add-on package location>
```

Example:

```
> ./<CM software location>/bin/addon-software-install -p fms_ui_8.0.x.x.jar
```

In this example, we are deploying the add-on software package that is located in the same directory where we are executing this command.

3. Listing available instance types of an add-on

To list all available instances types provided by the add-on, you have to execute the next line:

```
> ./<CM software location>/bin/config-generator -a <add-on name> -s
```

Example:

```
> ./<CM software location>/bin/config-generator -a fms_ui -s
```

The output result is something like this:

```
Copyright (c) WeDo Technologies - Assuring your business for the future
All Rights Reserved

Available Instance Types:
    -> "FMS_UI"
```

4. Generate and adjust configuration file

Since add-ons may require additional configuration values, you must generate a new configuration file for the add-on instance type.

To generate the configuration file, execute the following command:

```
> ./<CM software location>/bin/config-generator -a <add-on name> -t <add-on
instance type> -o <output file>
```

Example:

```
> ./<CM software location>/bin/config-generator -a fms_ui -t FMS_UI -o config.xml
```

Note: After the file generation, you have to manually merge the new configuration file with the one you were using previously for CM, ensuring that no configuration values are discarded.

Check the Case Management Installation Manual  [CM34] if you need to validate CM parameters.

5. Install add-on in an existing instance

This step installs the add-on into an existing instance. To install this add-on for multiple instances, you must repeat this step for each instance. ***This step is only applicable to products with instances installed.***

```
> ./<CM software location>/bin/addon-instance-install -a <add-on name> -t <add-on
instance type> -n <instance name> -c <configuration file> -l <licence file>
```

Example:

```
> ./<CM software location>/bin/addon-instance-install -a fms_ui -t FMS_UI -n CM -
c config.xml -l raidfms.lic
```

This example installs the instance type **FMS_UI** of **fms_ui** add-on in the **CM** instance using the **config.xml** file to supply the configuration values.

6. Start instance server(s)

The final step is to restart the server and verify that the functionalities provided by the add-on are available. ***This step is only applicable to products with instances installed.***

```
> ./<CM software location>/instances/<instance name>/bin/start
```

9 How to Install GUIs

RAID:FMS is composed of a set of modules that work in an integrated environment consisted of the following modules:

- AB – Application Browser;
- BCM (Business Concepts Manager) – Modulation of data schemas;
- BPM (Business Process Manager) – Workflows design and management;
- CMF (Connection Manager Factory) – Connection manager;
- CSM – Context Search Manager;
- DASHBOARD – Graphic reports;
- EH – Event Handler;
- IM (Integration Module) – Data loading;
- IUD - Insert Update Delete tasks;
- RAID – Core;
- RAID:FMS;
- READERS - Additional IM Readers;
- RM (Report Module) – Reporting;
- SEARCH - File search.

To install RAID:FMS, execute the installation file:

RAID_FRAUD_8.0.x.x.exe

If the program is already installed, the uninstall process starts as shown in the next three figures. When the uninstall process starts, it automatically detects RAID's installation folder. To begin the uninstall process, simply click **Next**.

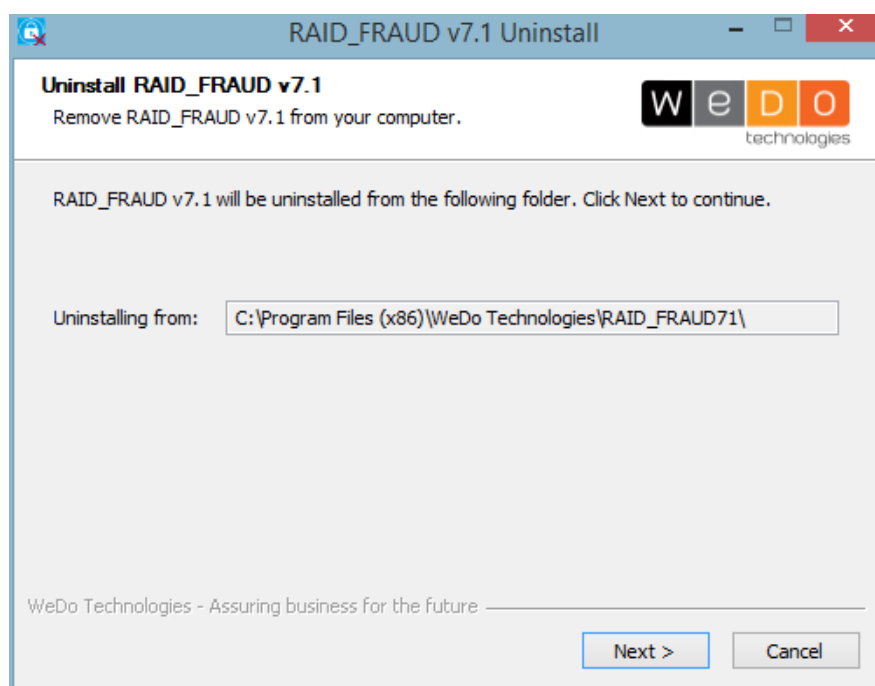


Figure 6 – RAID:FMS uninstall

At this point, the uninstall process asks you which components you want to uninstall. By default, all components are selected. To continue, select the components to uninstall and click the **Uninstall** button.

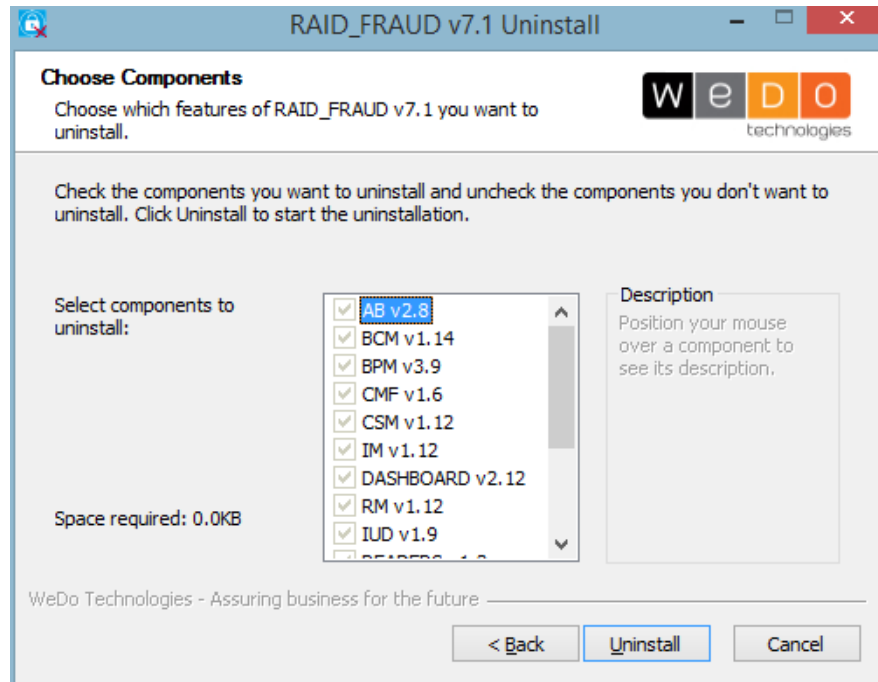


Figure 7 – Select components to uninstall

When concluded, click **Close** to finish the uninstall process.

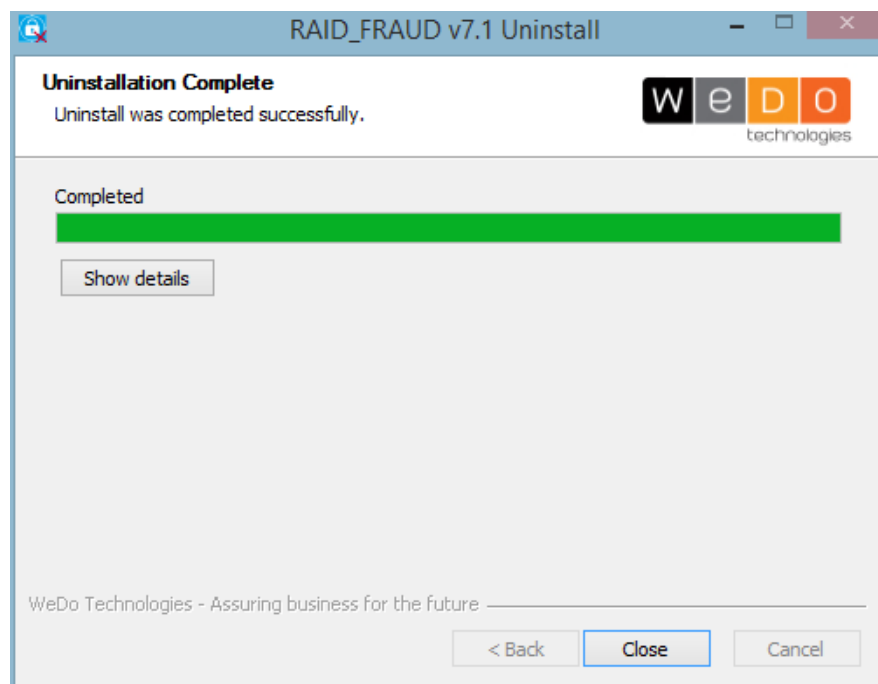


Figure 8 – Uninstall process conclusion

After the uninstall process is completed, you should run the installation command again:

RAID_FRAUD_8.0.x.x.exe

The installation process begins with the following window:

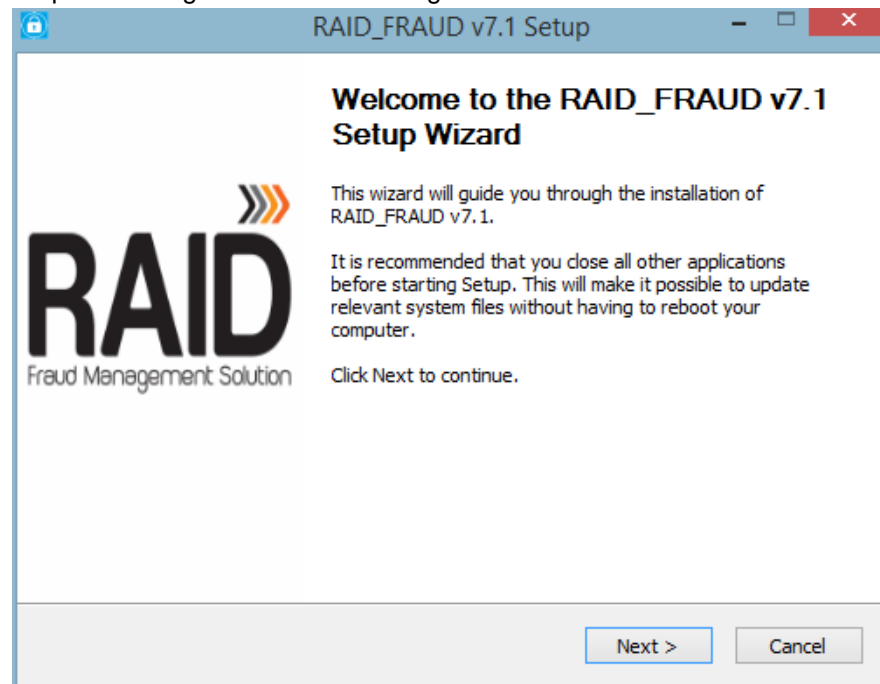


Figure 9 – RAID:FMS installation

Next, you are asked to accept the licensing agreement. You must accept it by clicking on the **Next** button continue to the next step.

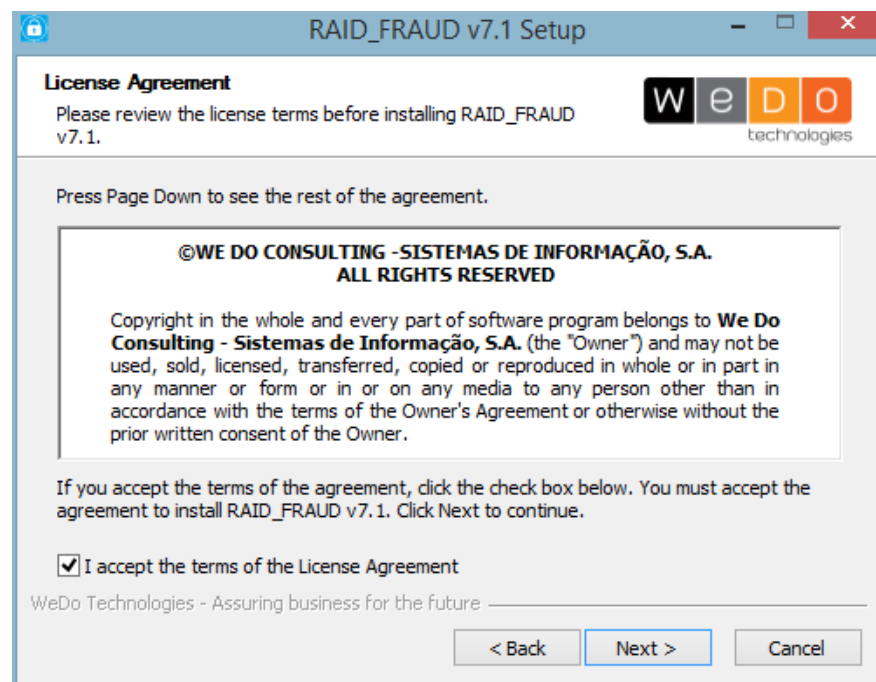


Figure 10 – License agreement terms

Select the components you want to install by checking the box next to each available component. After the components' selection, click **Next** to continue the installation process.

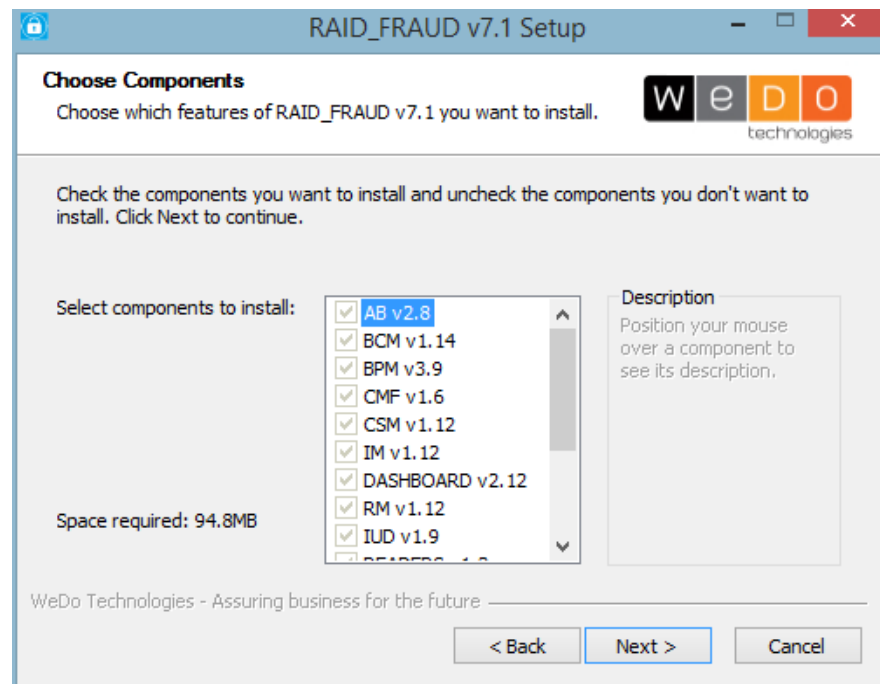


Figure 11 – Selection of the components to install

Choose the installation directory and click **Install**. A destination folder is suggested by default.

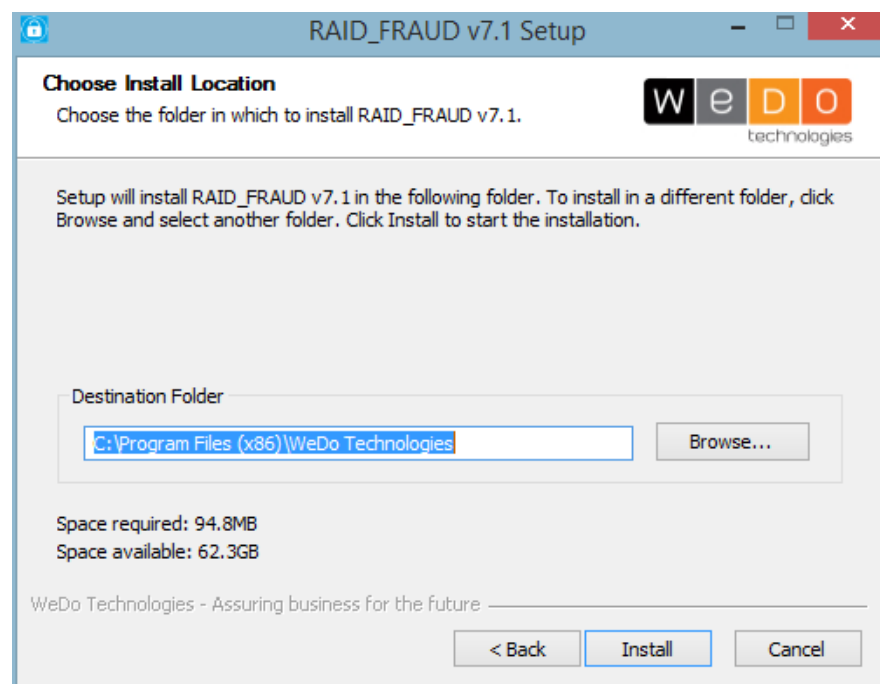


Figure 12 – Installation directory

When the installation process is completed, click **Close** to finish.

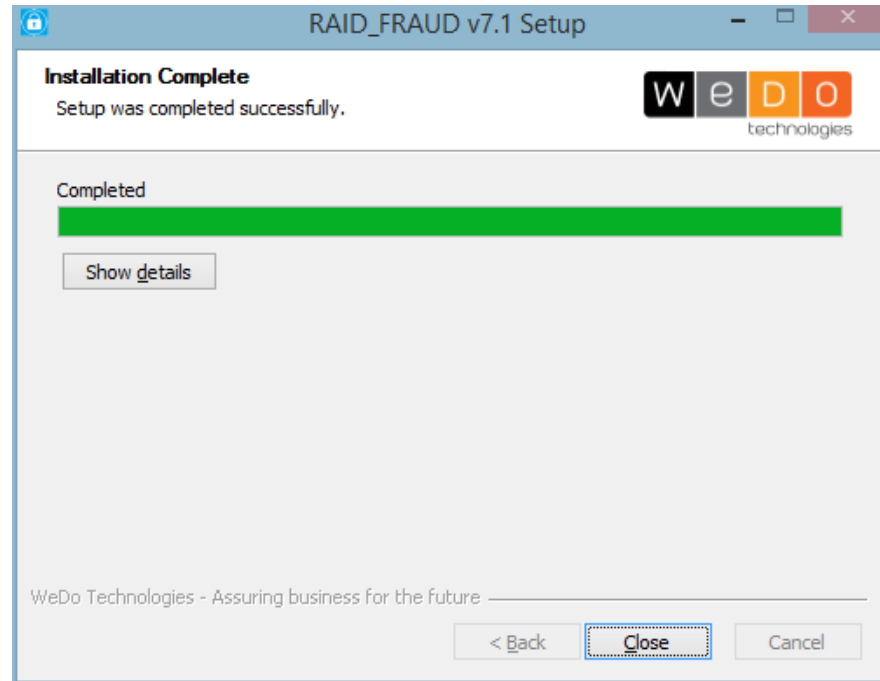


Figure 13 – Installation process conclusion

The installation can also be done silently using the following command line:

```
RAID_FRAUD_8.0.x.x.exe /S
```

With this type of installation no human interaction is needed and all the parameters are used with the defaults.

This can also be used to update the installation or to uninstall it in the same manner.

After a successful installation, place your license file `License.dat` in the `AB28` directory, next to the `AppBrowser` application.

Also, edit the `appbrowser.properties` file under the `AB28\config` directory to include the names and URLs of the server that are accessed. In this particular case, you should add the following line in the `appbrowser.properties`. Please note that the portal URI is `/xdt/`.

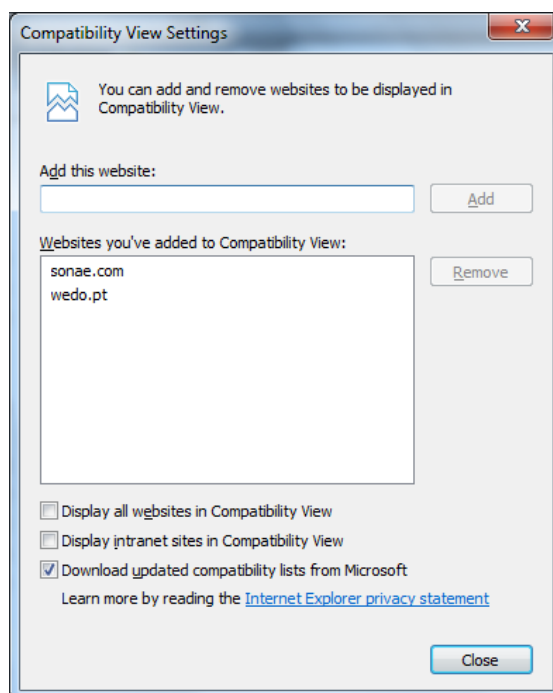
```
....
;#####
[Servers]
;-----
;AF servers URL as <aliasname> = (http|https)://<computer>:<port>/<uri>
;-----
RaidFMS-Prod = http://brwhp01:51019/xdt/
```

10 Troubleshooting

10.1 IE Settings

Due to the fact that IE has a fallback to compatibility mode when used on an intranet, the following options can be used:

- Use a FQDN (fully qualified domain name) to force IE to consider the URL as outside the intranet zone;
- Disable in all client browsers the fallback to compatibility mode:
 - Option “Display intranet sites in Compatibility View” should be disabled;
 - Option “Display all websites in Compatibility View” should be disabled.



10.2 Error Starting Component SEARCH

SEARCH requires several non-Java, system-dependent libraries.

During the SEARCH server start-up, a procedure is triggered to automatically detect which native libraries can be used for the current machine. If it fails, the following messages appear on the console where the server was launched:

```
*****
WARNING
Could not detect SEARCH Core libraries to use
Must be manually defined in ${INSTALL_DIR}/bin/.search-env-detect
*****
```


To fix the problem, the library must be set manually by editing the file mentioned in the first message. In this file, there is a commented section indicated by the value ***** Override here *****, containing a set of variables that must be set according to the sections that are described below.

After finishing this procedure, the server must be restarted.

10.3 Error Starting Component IM.Core

This component requires a non-Java, system-dependent library – IMDumper.

During the Backend server startup, a procedure is triggered to automatically detect which IMDumper library version can be used for the current machine. If it fails, the following message appears on the console where the server was launched:

```
*****
WARNING
Could not detect IMDumper library to use
Must be manually defined in
<WEDOPORTAL_BASE_DIR>/instances/<INSTANCE_NAME>/bin/.pre_start/native-lib.conf
*****
```

And the following identical messages appear in the log:

```
Failed to start service "<INSTANCE_NAME>.IM.Core" - Failed to load native
IMDumper library - no IMDumper in java.library.path
...
ERROR - AppMgr - Problems while launching applicationwedo.jaf.JafException:
Failed to launch "IM.Core" service - Failed to load native IMDumper library -
no IMDumper in java.library.path
...
WARN - AppMgr - wedo.jaf.JafException: Falhou o arranque do serviço "IM.Core"
- Failed to load native IMDumper library - no IM Dumper in java.library.path.
```

If the Backend server is running in Windows/Cygwin, this can be solved in two ways. You can either set the variable `ORACLE_HOME` to the home directory of Oracle or copy the appropriate `IMDumper.dll` file to the "windows\system32" directory. If you chose the second solution, the first warning message still appears but it should run properly.

On other systems, the library must be set manually by editing the file mentioned in the first message. In this file, there is a commented section indicated by the value ***** Override here *****, containing a set of variables that must be set according to the sections that are described below. After finishing this procedure, the server must be restarted.

10.4 Error Starting Component EVTTS

This component requires a non-Java, system-dependent library – EVTTS (Events Tracking and Summarization).

During the server startup, a procedure is triggered to automatically detect which EVTTS library version can be used for the current machine. If it fails, the following message appears on the console where the server was launched:

```
*****
WARNING
Could not detect EVTTS library to use
Must be manually defined in
<RAIDFMS_DEPLOY_DIR>/instances/<INSTANCE_NAME>/bin/.pre_start/evtts-native-
lib.conf
.....
```

And the following identical messages appear in the log:

```
Failed to start service "<INSTANCE_NAME>.EVTTS.Core" - Failed to load native
EVTTS library - no EVTTS in java.library.path
...
ERROR - AppMgr - Problems while launching applicationwedo.jaf.JafException:
Failed to launch "EVTTS.Core" service - Failed to load native EVTTS library -
no EVTTS in java.library.path
```

The library must be set manually by editing the file mentioned in the first message. In this file, there is a commented section indicated by the value ***** Override here *****, containing a set of variables that must be set according to the sections that are described below. After finishing this procedure, the server must be restarted.

10.5 EVTTS Library

The RAID:FMS is bundled with several versions of this library that are located under `<RAIDFMS_DEPLOY_DIR>/software/components/evtts/lib` directory. All bundle versions are organized upon a directories structure that follows the following pattern:

```
<Operating System>/<Machine Architecture>/<JVM Model>b/oracle<Oracle Client Version>
```

The installation process automatically selects the proper directory `hp-ux/ia64/64b/oracle11`, etc., depending on the operating system, machine architecture, Oracle Client and Java Virtual Machine model. After this procedure, the library version is dynamically loaded at the server startup.

To manually select the proper version, you must consider the following:

- The version being used must match the Oracle Client version that `ORACLE_HOME` environment variable points to;
- The number of bits of the selected library must match those of the JVM and also the `$ORACLE_HOME/lib` (lib32 or lib64) that the variable `LD_LIBRARY_PATH` or `SHLIB_PATH` (depending on the machine type) points to;
- The library file must have execution permission for the user account owner of the RAIDFMS server.

10.6 Error Rendering Web Content

Some rendering problems were reported by web client side users that are related with the use of `msxml3.dll`. The problems exist if, for some reason this dll is unregistered.

An example of a rendering error of this type is presented in the following image:

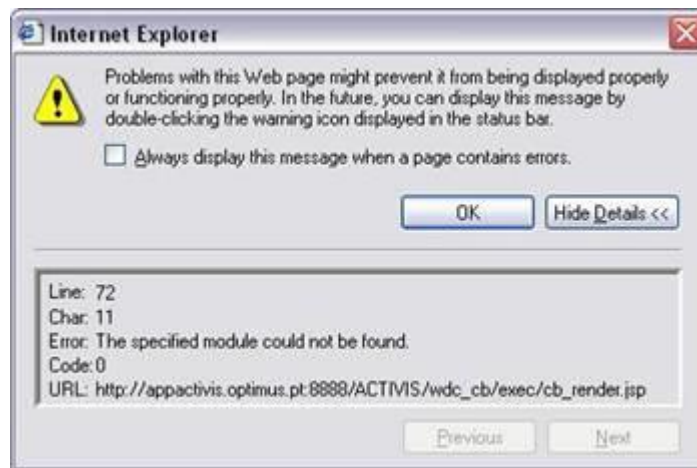


Figure 14 – Web client side rendering error

In order to fix the problem, you must manually register `msxml3.dll`.

To do so, open Start menu / Run ... and type

regsvr32 C:\WINDOWS\system32\msxml3.dll

followed by **Ok**.

A system message is shown confirming the registration success.

11 Appendix

11.1 Recovering a Failed Installation

Sometimes the installation/update process can fail due to unexpected errors. This section describes a methodology for recovering from those errors either by rolling back to the last known configuration or by fixing the error reasons. Some of these procedures require database and filesystem backup prior to installation.

The RAID:FMS installation provides a mechanism for bypassing unit installation/updates. This mechanism is useful when the installation is stuck on a given unit.

11.1.1 Rolling Back Installation

During the installation process, the following changes occur:

- Database changes by executing DML and DDL statements;
- Filesystem changes;
- Software and Instance Registry changes.

If the installation process stops at a point where it is not executing a DDL, it should be able to continue normally when the error is solved (if the error is external to the application). Common errors that prevent installation from being completed are:

- Environment Changes – Changes performed manually on the database which are now being sent in this update;
- Database related errors - Tablespaces have no free space, database unavailability;
- Filesystem errors - Filesystem is full.

After fixing those errors manually, the `update-instance` command can be re-issued in order to continue with the installation.

If the error cannot be solved, you need to perform backup recovery by:

- Restoring old directory contents including software and instance registry;
- Restoring database content using:
 - A previously saved export for the list of provided tables;
 - Full Database Recovery: From an Oracle Restore Point created at the beginning of an installation and using the FLASHBACK DATABASE feature;
 - From an Oracle Restore Point created at the beginning of an installation and using FLASHBACK TABLE feature. In this case, all DDL statements must be rolled back manually by performing the reverse DDL operation.

11.1.1.1 Full Database Recovery

This section describes the steps involved with full database recovery using Oracle Flashback database technology.

STEP 1: Database instance is shutdown.

```
SQL> shutdown immediate;

Database closed.

Database dismounted.

ORACLE instance shut down.
```

STEP 2: Database is flashbacked to saved restore point.

```
SQL> STARTUP MOUNT

ORACLE instance started.


Total System Global Area  209715200 bytes
Fixed Size                  1248140 bytes
Variable Size               67110004 bytes
Database Buffers           138412032 bytes
Redo Buffers                2945024 bytes
Database mounted.

SQL> FLASHBACK DATABASE TO RESTORE POINT before_install ;

Flashback complete.
```

STEP 3: Database is open, restore point is dropped.

```
SQL> ALTER DATABASE OPEN RESETLOGS;

Database altered.

SQL> drop restore point testel;

Restore point dropped.
```

11.1.1.2 Table-based Recovery

This recovery mode allows you to undo all DML statements performed during installation/upgrade.

Note: you must disable any database activity besides the installation procedure to prevent data from being lost.

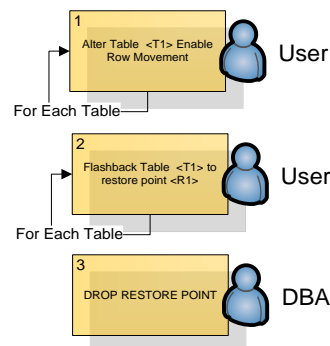


Figure 15 – Table-based recovery

STEP 1: Enable `ROW MOVEMENT` in every table where data is being recovered. If the table already has `ROW MOVEMENT` enabled, skip this step.

```
SQL> ALTER TABLE <table_name> ENABLE ROW MOVEMENT;
```

STEP 2: Perform flashback recovery on tables. This operation only undoes DML changes and not DDL changes.

```
SQL> FLASHBACK TABLE <table_name> TO RESTORE POINT before_install;
```

STEP 3: Finally, drop the restore point. If this was a `FLASHBACK DATABASE` restore point, only the DBA is able to drop it.

```
SQL> DROP RESTORE POINT before_install;
```

11.2 Backup

In order to ensure point-in-time recovery, it is necessary to ensure the appropriate backup level. In this section, database and filesystem backup is covered.

11.2.1 Database Backup

Prior to ORACLE 10, only export/import tools were available, and these were the suggested methods for backing up instance data. In ORACLE10 or greater, the use of flashback technologies is advised because it is faster and easier to use, however some limitations still do exist.

This section describes the steps involving full point-in-time recovery of the database instance. The following steps need to be performed before the update procedure is run, and ensures complete rollback of the changes performed during the update procedure. Due to the nature of the task, a user with SYSDBA role is required.

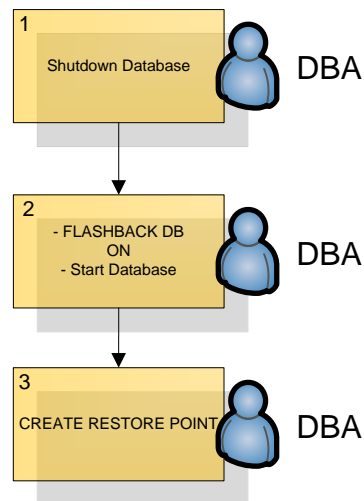


Figure 16 – Full point-in-time recovery

STEP 1: Instance is shutdown.

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
```

STEP 2: Database instance is mounted and database support for flashback is enabled (flashback area should already be initialized at this step).

```
SQL> startup mount;
ORACLE instance started.

Total System Global Area  209715200 bytes
Fixed Size                  1248140 bytes
Variable Size              62915700 bytes
Database Buffers          142606336 bytes
Redo Buffers                2945024 bytes
Database mounted.

SQL> ALTER DATABASE FLASHBACK ON;

Database altered.

SQL> ALTER DATABASE OPEN;

Database altered.
```

STEP 3: A restore point is created enabling point-in-time recovery to the current timestamp.

```
CREATE RESTORE POINT before_install GUARANTEE FLASHBACK DATABASE;
```

After this last step, the system tracks all changes and is able to reset database to the current snapshot. Be aware that all database activity should be restricted to RAID during the intervention because if a `FLASHBACK DATABASE` command is issued, all schemas return to the saved snapshot.

11.2.2 Filesystem Backup

With all instances stopped, create a backup copy of the installation folder synchronized with the database snapshot. Please ensure that the backup directory is empty before continuing.

```
>mkdir -p <deploy_dir backup>  
>cp -R <deploy_dir> <deploy_dir backup>
```


Please, give us your opinion about:

Manuals	Product
Please let us know if this manual fits your needs, and if there are areas where it can be improved.	Please let us know if there are features that you would like see in the product or others that should be improved.
Email: customerservices@wedotechnologies.com	Email: customerservices@wedotechnologies.com