Installation Guide

entellitrak Application Manager

Version 1.0

PUBLISHED BY

MicroPact, Inc. 12901 Worldgate Drive, Suite 800, Herndon, VA 20170

COPYRIGHT © MICROPACT, INC. All rights reserved.

No part of the contents of this manual may be reproduced or transmitted in any form or by any means without written permission.

entellitrak-h-RGBis a registered trademark of MicroPact, Inc. Other product and company names mentioned herein may be the trademarks of their respective owners.

entellitrak is a data tracking and management platform that can be implemented immediately and configured continuously to model your workflow and business processes with minimal programming. Whether it's based on-premises or in the cloud, entellitrak's streamlined approach scales from the local office to the enterprise, giving everyone in your organization the ability to accurately capture, track and report on the data that drives operations forward.

We welcome feedback from our users regarding the entellitrak product or documentation. If you have comments, corrections, suggestions, etc. with regard to the product or this manual please email: support@entellitrak.com

SUPPORT INFORMATION

🕿 DIRECT 703.709.6110

🕿 TOLL FREE 1.866.346.9492

🖨 FAX 703.709.6118

🖂 EMAIL: [support@entellitrak.com](mailto:support@entellitrak.com)

🖳 ONLINE: [http://www.entellitrak.com](http://www.entellitrak.com/)

Contents

[1 Overview 1](#_Toc428456018)

[1.1 System Requirements 1](#_Toc428456019)

[1.2 Prerequisites 2](#_Toc428456020)

[2 Installation 3](#_Toc428456021)

[2.1 Configuring the eam.properties File 3](#_Toc428456022)

[2.2 Configuring the dsConfig.xml File 4](#_Toc428456023)

[3 Starting entellitrak Application Manager 6](#_Toc428456024)

[3.1 Using eAM with External Tomcat Server 6](#_Toc428456025)

[4 Temporary Folder Configuration 8](#_Toc428456026)

[4.1 Operations Maintenance Requirements 8](#_Toc428456027)

[5 Troubleshooting Application Startup Errors 9](#_Toc428456028)

[6 Sample Files 10](#_Toc428456029)

[6.1 eam.properties configuration file 10](#_Toc428456030)

[6.2 Sample dsConfig.xml configuration file 11](#_Toc428456031)

List of Figures

[Figure 3‑1: Tomcat Properties 6](#_Toc428456017)

List of Tables

[Table 2‑1: eam.Properties File 6](#_Toc428453274)

[Table 2‑3: dsConfig.xml File 8](#_Toc428453275)

# Overview

entellitrak Application Manager (eAM) is a standalone application that works with entellitrak systems to move entellitrak configuration data between applications. Configuration data from a site can be deployed to an application bundle, or a file archive of data. Application bundles can then be deployed to other applications. For example, an application bundle can be created from a development application/entellitrak site, then deployed to a new production application/entellitrak site. The file structure of application bundles allow them to be easily distributed, compared (for example, by using a “diff” tool), and used with a version control system.

This document describes how to install and configure eAM for use with entellitrak applications.

## System Requirements

The following are the system requirements for eAM:

* Operating system
  + Windows
  + Linux
* Databases:
  + Oracle 11g
  + Microsoft SQL Server 2008, and 2012
* Java
  + Oracle Java 7 Standard or Enterprise Edition
* Application Server
  + Tomcat 7.x
* Memory Requirements:
  + Minimum 512 MB of hard drive space
  + Minimum 512 MB of RAM

## Prerequisites

This document assumes that the following prerequisites have been met:

* The applications that will be managed by eAM must have entellitrak software version that is supported by the given version of eAM installed. The entellitrak Application Manager Guide contains a list of the entellitrak versions supported by each eAM version.
* A working Tomcat 7.x environment is installed as the application server
* JDBC driver for a given database vendor is in the runtime classpath (TOMCAT\_HOME\lib folder)
  + Oracle JDBC Driver version 11.2.0.3 recommended
  + SQL Server JDBC Driver version 4.0 recommended

# Installation

The eAM application is provided for installation in the form of a web application archive (WAR) file. The WAR must be deployed as a web application to an existing Tomcat server.

eAM requires configuration files to connect to entellitrak applications. These configuration files can be stored externally (not included in the WAR file), or embedded in the WAR file. The required files are described in the following sections.

## Configuring the eam.properties File

The eam.properties file specifies operating environment-level properties. A sample file is included in the WAR file, and is also included in the Sample Files section. The file must be modified with your system information (for example, by using a text editor to change the appropriate properties). The following table describes the properties that should be configured.

|  |  |  |
| --- | --- | --- |
| Property | Default Value | Description |
| spring.profiles.active | sqlserver | Based on the databases configured in the dsConfig.xml file, this should be set to either *oracle*, or *sqlserver*. |
| temp.file.path | C://eam\_temp | The location used for temporary storage of exported configuration data. Edit the property to reflect the location of the folder on your system (for example, *C:\eam\_temp*). See the Temporary Folder Configuration section below for more information. |
| multipart.max-file-size | 100MB | The maximum size of files that can be uploaded in the application. Accepts numbers followed by **KB** or **MB**. |

Table ‑: eam.Properties File

## Configuring the dsConfig.xml File

The dsConfig.xml file contains the database connection details needed for eAM to connect to entellitrak applications. The database connection details of the entellitrak applications to be managed by eAM must be configured in this file (for example, by using a text editor to change the appropriate properties). The following table describes the properties that should be configured; if a property is not listed, the default value is recommended and the property should not be changed.

Some properties are configured based on the database used (SQL Server or Oracle). **All the configured data sources must use the same database** (for example, you cannot have a Development SQL server database and a QA Oracle database).

A sample file is included in the WAR file, and is also included in the Sample Files section.

|  |  |  |
| --- | --- | --- |
| Property | Default Value | Description |
| Data Source Information | | |
| Parent Data Source | (SQL Server):  <bean id="devDataSource" parent="**parentSQLServerDataSource**">  <bean id="qaDataSource" parent="**parentSQLServerDataSource**"> | If using Oracle, change to:  <bean id="devDataSource" parent="**parentOracleDataSource**">  <bean id="qaDataSource" parent="**parentOracleDataSource**"> |
| url | (SQL server example):  <property name="url" value="**jdbc:sqlserver://localhost:1433**;databaseName=devdb"/>  <property name="url" value="**jdbc:sqlserver://localhost:1433**;databaseName=qadb"/> | Change the default value to the URLs of the entellitrak applications to which eAM will connect. You must configure at least one data source, and there is no limit to the number of data sources that can be configured. For Oracle, the sample URL string is “jdbc:oracle:thin:@localhost:1521:XE” |
| Username | user | The database username. |
| password | password | The database password. |
| Key Mapping Information  Note: Keys must be unique. | | |
| entry key | Dev site  QA site | A user-friendly name for the data source that will be displayed in the drop-down list in eAM (for example, in the choose source and destination fields). Note that the key values should be unique. |
| value-ref | devDataSource  qaDataSource | The property name of the data source that is associated with entry key property. |
| defaultTargetDataSource | devDataSource | eAM uses the data source defined in this property to initialize the system. Using a development data source is recommended. |

Table ‑: dsConfig.xml File

# Starting entellitrak Application Manager

In the scenario described below, eAM uses external configuration files with an external Tomcat server. Note the following:

* These instructions assume that the external configuration files (the eam.properties file and the dsConfig.xml file) are located in the *C:\eAMConfig* folder.
* When eAM starts up, it will create a new table in all the entellitrak databases configured in the dsConfig.xml, if the table does not already exist. All the configured databases must be running before starting eAM.

## Using eAM with External Tomcat Server

In this example, Tomcat server is running as a service named *Tomcat7-8190*; use the name of your Tomcat service.

1. Verify that the JDBC driver is in the Tomcat’s *lib* folder.
2. From the command prompt, run the Tomcat Service Manager under TOMCAT\_HOME\bin:

**tomcat7w //ES//Tomcat7-8190**

1. Append the eAMConfig folder to the Java classpath, as shown in Figure 3‑1.
2. Add the following under Java Options, as shown in Figure 3‑1:

**-Dspring.config.location=file:/C:/eAMConfig/eam.properties**

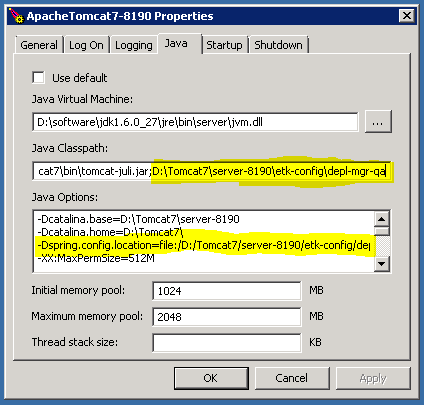


Figure ‑: Tomcat Properties

If the system is running Tomcat server manually using the exploded zip file, update the **setenv.bat** file in the *bin* folder:

1. Set CATALINA\_OPTS=-Dspring.config.location=file:/C:/eAMConfig/eam.properties
2. Add the following to CLASSPATH:

**SET "CLASSPATH=%CLASSPATH%;C:/eAMConfig”**

**Note**: The disadvantage of this approach is that only one eAM instance can exist on a given Tomcat server.

After starting eAM, access the application in your browser from the following URL:

**http://hostname:*portnumber*/*appname*** (where *hostname* is the name of the server running Tomcat, *portnumber* is the port that Tomcat is running on, and *appname* is the filename of the eAM application). For example: [**http://localhost:8190/eam**](http://localhost:8190/eam)

**Note**: You will have to restart the Tomcat server after making the above changes, if it is already running.

# Temporary Folder Configuration

eAM requires a temporary folder which is used to temporarily store files while they are being used by the application. The **temp.file.path** parameter is configured in the eam.properties file, as described in Table 1.

After eAM is configured, the temporary file path specified in the eam.properties file must also be created on the host operating system. The requirements for this folder are as follows:

* The path specified in the eam.properties file must exist on the host system.
* The user account that will be used to run eAM must have read and write rights to this folder (along with the rights to create files and sub-folders if the OS restricts that).
* There must be sufficient disk space allocated to the path specified for eAM operations. This size will vary depending on the usage of the eAM and the size of the entellitrak applications that it manages. Use at least 1GB of available space and take adequate maintenance precautions to prevent space from running out.

## Operations Maintenance Requirements

The temporary folder used by eAM must be monitored by the system administrator for the duration of eAM use. This folder creates and expands .eab files (entellitrak application bundles) that are used in the eAM workflow. eAM does perform maintenance on this folder and its files, but in some cases, old data can remain on the server. Perform a regularly-scheduled maintenance operation on this folder to prevent the disk becoming full or collisions with folder names. We recommend creating a system maintenance script that runs every week and clears the temporary folder of any files or folders with creation dates older than a week.

# Troubleshooting Application Startup Errors

For eAM to start up without any errors, the *dsConfig.xml* & *eam.properties* files must exist, must be valid, and contain all the required properties. If any error occur due to missing/ invalid configuration files, the Tomcat logs should contain the cause for the startup failure. Some of the common issues that occur when deploying the application are listed below.

* A *SAXParseException*/ *XmlBeanDefinitionStoreException* in the stack trace is an indication of an XML syntax error in the *dsConfig.xml* file. Usually the line number containing the error will be included in the stack trace. For example:

*XmlBeanDefinitionStoreException: Line 54 in XML document from class path resource [dsConfig.xml] is invalid; nested exception is org.xml.sax.SAXParseException; lineNumber: 54; columnNumber: 11;*

* If data sources are not defined, or are incorrectly defined in *dsConfig.xml* file, an error message similar to one of the following should be seen in the logs:
  + *BeanCreationException: Error creating bean with name 'dataSource' defined in class path resource [dsConfig.xml]: Cannot resolve reference to bean 'devDataSource' while setting bean property 'defaultTargetDataSource'; nested exception is org.springframework.beans.factory.NoSuchBeanDefinitionException: No bean named 'devDataSource' is defined*
  + *BeanCreationException: Error creating bean with name 'dataSource' defined in class path resource [dsConfig.xml]: Cannot resolve reference to bean 'devDataSource' while setting bean property 'targetDataSources' with key [TypedStringValue: value [Local Dev]; nested exception is org.springframework.beans.factory.NoSuchBeanDefinitionException: No bean named 'devDataSource' is defined*
* The application will also fail to startup if it cannot connect to either of the configured data sources. A sample error message if a SQL Server database is unreachable is as follows:

*ERROR o.s.boot.SpringApplication - Application startup failed; nested exception is com.microsoft.sqlserver.jdbc.SQLServerException: The TCP/IP connection to the host localhost, port 1433 has failed. Error: "Connection refused: connect. Verify the connection properties. Make sure that an instance of SQL Server is running on the host and accepting TCP/IP connections at the port. Make sure that TCP connections to the port are not blocked by a firewall.*

* The application will fail to startup if the appropriate JDBC driver is not in the *classpath*. For example, if JDBC driver for SQL Server is missing, the error message will look like:   
    
  IllegalStateException: Could not load JDBC driver class [com.microsoft.sqlserver.jdbc.SQLServerDriver]
* The application will fail to startup if the active profile *(spring.profiles.active)* value is missing or incorrect in the *eam.properties* file. The following error message can be seen in the logs:

IllegalStateException: Unable to find any schema resource files. Please verify the active profile name configured in eam.properties file.

# Sample Files

## eam.properties configuration file

# ===========================================================================================

# This is a sample configuration file for Application Manager.

# You can copy and paste these settings to "eam.properties" file and modify them accordingly.

# ===========================================================================================

# Pick only one of the following profiles, based on your databases in dsConfig.xml file: oracle, sqlserver

spring.profiles.active = sqlserver

# The temporary folder to store eAM files

temp.file.path = C://eam\_temp

# The maximum size for files which can be uploaded in the application.

# Accepts numbers followed by "KB" or "MB".

multipart.max-file-size = 100MB

## Sample dsConfig.xml configuration file

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:security="http://www.springframework.org/schema/security"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-4.0.xsd">

<!-- Parent data source for SQL Server database. DO NOT EDIT -->

<bean id="parentSQLServerDataSource"

class="org.springframework.jdbc.datasource.DriverManagerDataSource"

abstract="true">

<property name="driverClassName" value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />

</bean>

<!-- Parent data source for Oracle database. DO NOT EDIT -->

<bean id="parentOracleDataSource"

class="org.springframework.jdbc.datasource.DriverManagerDataSource"

abstract="true">

<property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />

</bean>

<!-- Define one or more data sources of the same parent type. -->

<!-- Use the parent attribute depending on Oracle/ SQL Server -->

<!-- If Oracle, use parent="parentOracleDataSource" -->

<!-- If SQL Server, use parent="parentSQLServerDataSource" -->

<!-- Note that you can use only one of the two parent's,

i.e. the configuration should contain either Oracle/ SQLServer data source, and NOT both. -->

<bean id="devDataSource" parent="parentSQLServerDataSource">

<property name="url" value="jdbc:sqlserver://localhost:1433;databaseName=devdb"/>

<property name="username" value="user"/>

<property name="password" value="password"/>

</bean>

<bean id="qaDataSource" parent="parentSQLServerDataSource">

<property name="url" value="jdbc:sqlserver://localhost:1433;databaseName=qadb"/>

<property name="username" value="user"/>

<property name="password" value="password"/>

</bean>

<!-- Define the key mappings for different data sources, and select the default data source. -->

<!-- Note that the keys will be used in the UI to display available data sources to the user. -->

<!-- Keys must be unique -->

<bean id="dataSource" class="com.micropact.entellitrak.tenant.CustomRoutingDataSource">

<property name="targetDataSources">

<map key-type="java.lang.String">

<entry key="Dev site" value-ref="devDataSource"/>

<entry key="QA site" value-ref="qaDataSource"/>

</map>

</property>

<!-- Must pick one of the data sources as the default data source -->

<property name="defaultTargetDataSource" ref="devDataSource"/>

<!-- Set to false to reject non-existent data sources connections. DO NOT EDIT -->

<property name="lenientFallback" value="false"/>

</bean>

</beans>

|  |  |  |  |
| --- | --- | --- | --- |
| Revision History | | | |
| Version Number | Date | Author | Description |
| 1.0 | 08/27/2015 | Product Team/Product Information Team | Initial draft. |
|  |  | Product Team/Product Information Team |  |