

1. BellaDati Developers Network	2
1.1 System requirements	2
1.2 Supported platforms	5
1.3 Installation and upgrade guide	6
1.3.1 Installing JAVA	7
1.3.2 Installing BellaDati WAR	7
1.3.2.1 Installing BellaDati on Glassfish 2.1	8
1.3.2.2 Installing BellaDati on Jetty 6	13
1.3.2.3 Installing BellaDati on Tomcat 6	15
1.3.2.4 BellaDati WAR Configuration Overview	19
1.3.3 Installing BellaDati virtual appliance	22
1.3.3.1 Installing BellaDati on VMWare ESX and ESXi	23
1.3.3.2 Installing BellaDati on VMWare Player	26
1.3.3.3 Launching BellaDati virtual appliance	28
1.3.4 Setup BellaDati data warehouse	30
1.3.4.1 Setup BellaDati with PostgreSQL	31
1.3.5 Running the setup wizard	33
1.3.6 Upgrading BellaDati	35
1.3.6.1 Upgrading BellaDati virtual appliance	35
1.3.6.2 Upgrading BellaDati WAR	37

BellaDati Developers Network

Installation Guide

The [BellaDati Installation Guide](#) is for people who are installing BellaDati on-premise for the first time. Check the [requirements](#) and [supported platforms](#), then [download](#) and install BellaDati.

Developer Resources

These resources are for software developers who want to create their own applications using the BellaDati API or prepare scripts for advanced data transformation. Take a look at the [Developer Tools](#). You may also find the BellaDati Developers Forum useful.

Upgrade Guide

The [BellaDati Upgrade Guide](#) is for people who are upgrading their instance of BellaDati on-premise. Start by reading the latest [Release Notes](#) for the version to which you are upgrading, then [download](#) BellaDati and follow the main Upgrade Guide.

System requirements



BellaDati is a web application, meaning it runs centrally on a server, and users interact with it through web browsers from any computer.

BellaDati Client/Server Software Requirements

Please read the [Supported platforms](#) page for BellaDati. The BellaDati Supported Platforms page contains important information on the client/server software supported for, including:

- Web browsers
- Operating systems (if you are installing [BellaDati WAR](#))
- [JAVA](#) (if you are installing [BellaDati WAR](#))
- Application servers (if you are installing [BellaDati WAR](#))
- Databases (if you are installing [BellaDati WAR](#))
- Virtual servers (if you are installing [BellaDati virtual appliance](#))

Please also read the information below regarding client/server software for BellaDati.

Browser requirements

PC users

- Google Chrome
- Mozilla Firefox 3+
- Internet Explorer 8+

MAC users

- Mozilla Firefox 3+
- Safari 5+

Other requirements

- enabled Javascript and cookies
- installed Adobe Flash 9+
- display with min. 1024px(recommended 1280px) resolution
- internet or intranet access (applicable to on-premise version only) with speed at least 1024/256mbps (download/upload)
- appropriate application for viewing exported files (PDF, XLS , PPT, PNG)

On this page:

- [BellaDati Client/Server Software Requirements](#)
 - [Browser requirements](#)
 - [JAVA platform requirements](#)
 - [Application Server](#)
 - [Database Server](#)
- [BellaDati Server Hardware Recommendations](#)
 - [Minimal configuration \(single user is analyzing thousands records\)](#)
 - [Medium configuration \(tens of users are analyzing hundred thousands records\)](#)
 - [Application server](#)
 - [Data warehouse server](#)
 - [High loaded server configuration \(hundreds of users are analyzing millions records\)](#)
 - [Application server](#)
 - [Data warehouse server](#)
- [Next Steps](#)

JAVA platform requirements

BellaDati requires Oracle's (formerly Sun's) Java Development Kit (JDK) platform to run. Refer to [Supported platforms](#) for details on the Java platform versions that BellaDati supports. For instructions on how to do it, please refer to [Installing Java](#).



Currently Oracle's JDK is available for Windows (32 bit + 64 bit), Solaris (SPARC + 32 bit + 64 bit) and Linux Platforms (32 bit + 64 bit). If you are a Mac OSX user, Apple also provides a specific JDK that has been optimised for its hardware and OS.

Application Server

BellaDati is a web application, so it requires an application server. BellaDati WAR can be installed into any of the supported application servers (see [Supported platforms](#)) provided they are compatible with your chosen operating system and JDK. Note that manual configuration will be required.

Database Server

BellaDati requires a relational database to build its data warehouse.

BellaDati Server Hardware Recommendations

During evaluation, BellaDati will run well on any reasonably fast workstation computer (e.g. 1.5+ Ghz processor). Memory requirements depend on how many data you will store and analyze, but 2048MB (of Java heap size) is enough for most evaluation purposes.

Most people start by installing BellaDati virtual appliance for VMWare player or BellaDati WAR on their local computer and migrate this to a production server later.

The hardware required to run BellaDati in production depends mainly on data amount that your installation will process, as well as the maximum number of concurrent requests that the system will experience during peak hours. In most cases, BellaDati can handle user requests in a few seconds. But there are many extreme situations, which is necessary to be reflected in the hardware configuration. For this reason, the hardware requirements are divided into three categories:

Minimal configuration (single user is analyzing thousands records)

Application server and database are running on the same server.

CPU	1x 1Ghz 32bit Intel Xeon (or equivalent)
-----	--

RAM	2GB	
Disk	2GB, 7200ot., IDE/SCSI	depends on data warehouse size
Network	10 Mbps	



This is the minimal configuration, which enables BellaDati to run. We do not recommend to use it for production purposes.

Medium configuration (tens of users are analyzing hundred thousands records)

Separated application and database server is strongly recommended for this configuration.

Application server

CPU	2x 2Ghz 64bit Intel Xeon (or equivalent)
RAM	4GB
Disk	4GB, 10000ot., IDE/SCSI (S-ATA/SAS)
Network	100 Mbps

Data warehouse server

CPU	2x 2Ghz 32bit Intel Xeon (or equivalent)	
RAM	4GB	
Disk space	10GB, 10000ot., RAID	depends on data warehouse size
Network	100 Mbps	

High loaded server configuration (hundreds of users are analyzing millions records)

Separated application and database server is strongly recommended for this configuration.

Application server

CPU	4x 2.4Ghz 64bit Intel Xeon (or equivalent)
RAM	12GB
Disk space	20GB, 10000ot., IDE/SCSI (S-ATA/SAS)
Network	1 Gbps

Data warehouse server

CPU	4x 2.4Ghz 64bit Intel Xeon (or equivalent)	
RAM	8GB	
Disk space	60GB, 15000ot., RAID	depends on data warehouse size
Network	1 Gbps	

Please note that performance heavily depends on your usage pattern.



We would appreciate it if you let us know what hardware configuration works for you. Please create a ticket in our [support system](#) with your hardware specification and mention the number of users and issues in your BellaDati installation.

Next Steps

[Installation and upgrade guide](#)

Supported platforms



This page describes the supported platforms for BellaDati 2.X.

OS Platforms

BellaDati is a pure web Java Enterprise application and run on any platform which supports Application server/container and JDK:

- Windows
- Linux (reference implementations for: RedHat, SUSE, CentOS)
- Solaris SPARC
- Solaris x86
- MacOS
- AIX



BellaDati is supported running on x86 hardware and 64-bit derivatives of x86 hardware.

Client side of BellaDati is platform independent. No client installation is required. User have to use compatible web browser with Flash.

JAVA Runtime environment

BellaDati requires Oracle's (formerly Sun's) Java Development Kit (JDK) platform to run. BellaDati supports Java SE 6 Update 17 or higher.

Oracle's JDK can be downloaded from [Oracle's website](#).



Linux distributions frequently have an open-source implementation of Java called GCJ installed. Do not use this Java platform — it is incomplete and BellaDati will not run successfully on it.

Application server

BellaDati application itself resides in WAR (web application archive). It has to be deployed into compatible Application server. Following application servers has been tested. Any other Java Application Server should be compatible.

Name	Supported version
GlassFish	2.1+
Jetty	6+
Apache Tomcat	6+

Notes

1. BellaDati is a pure Java application and should run on any platform if all the JDK requirements are satisfied.
2. Deploying multiple BellaDati applications in a single application container is not supported. We do not test this configuration and upgrading any of the applications (even for point releases) is likely to break it.

Database server

BellaDati creates own warehouse structure for storing data in a multi-dimesional cube. Following database servers has been tested.

Name	Version
PostgresSQL	8.4+
MySQL	5+
Oracle	10g+

Virtualised environment

BellaDati is distributed also as an virtual appliance, which enables for it to operate on these systems:

Name	Version
VMWare Hypervisor ESX	4.0+
VMWare Hypervisor ESXi	4.0+
VMWare Player	3+
VMWare Server	2.0+
Oracle VM VirtualBox	4.+
Xen Hypervisor	4.0+

Installation and upgrade guide

How to get BellaDati

Check the [BellaDati Download Page](#) for information about the current version and for downloading instructions.

This Installation Guide applies if you are installing BellaDati for the first time or if you are upgrading BellaDati.

Which 'Distribution' Should You Choose?

BellaDati is available in two 'distributions':


Virtual Appliance distribution	WAR distribution
Pre-packaged with the GlassFish application server	Deploys into an existing application server (GlassFish, Jetty, Tomcat).
Requires virtually no setup	Requires manual configuration
Recommended for VMWare users using the ESX/ESXi, VMWare Server or VMWare Player	Suitable when upgrading BellaDati

Next steps

The pages listed below contain information on installing and upgrading BellaDati:

- [Installing JAVA](#)
- [Installing BellaDati standalone](#)
 - [Installing BellaDati standalone on Windows](#)
- [Installing BellaDati WAR](#)
 - [Installing BellaDati on Glassfish 2.1](#)
 - [Installing BellaDati on Jetty 6](#)
 - [Installing BellaDati on Tomcat 6](#)
 - [BellaDati WAR Configuration Overview](#)
- [Installing BellaDati virtual appliance](#)
 - [Installing BellaDati on VMWare ESX and ESXi](#)
 - [Installing BellaDati on VMWare Player](#)
 - [Launching BellaDati virtual appliance](#)
- [Setup BellaDati data warehouse](#)
 - [Setup BellaDati with MySQL](#)
 - [Setup BellaDati with Oracle](#)
 - [Setup BellaDati with PostgreSQL](#)
- [Running the setup wizard](#)
- [Upgrading BellaDati](#)
 - [Upgrading BellaDati virtual appliance](#)
 - [Upgrading BellaDati WAR](#)


Installing JAVA

 You can skip this page if you are using [BellaDati virtual appliance](#)

1. Installing Java

BellaDati requires Oracle's (formerly Sun's) Java Development Kit (JDK) platform to run. Refer to [Supported Platforms](#) for details on the Java platform versions that BellaDati supports.

Oracle's JDK can be downloaded from [Oracle's website](#).

 Linux distributions frequently have an open-source implementation of Java called GCJ installed. Do not use this Java platform — it is incomplete and BellaDati will not run successfully on it.

You can test whether you have the correct Java platform by running `java -version`:

```
~$ java -version
java version "1.6.0"
Java(TM) SE Runtime Environment (build 1.6.0-b105)
Java HotSpot(TM) Client VM (build 1.6.0-b105, mixed mode, sharing)
```

On recent Linux distributions, Oracle's (formerly Sun's) JDK can be installed with a command like `sudo apt-get install sun-java6-jdk` (for Ubuntu).

2. Setting JAVA_HOME

Once the JDK is installed, you will need to set the `JAVA_HOME` environment variable, whose value is the root directory of the JDK.

Some JDK installers set this automatically (check by typing `echo %JAVA_HOME%` in a Windows command prompt, or `echo $JAVA_HOME` in a Linux/UNIX console).

Installing BellaDati WAR

To install BellaDati WAR, follow the instructions for your application server:

- [Installing BellaDati on Glassfish 2.1](#)
- [Installing BellaDati on Tomcat 6](#)
- [Installing BellaDati on Jetty 6](#)

 To install the BellaDati virtual appliance distribution, see [Installing BellaDati virtual appliance](#).

Additional Notes

- Read the [BellaDati WAR Configuration Overview](#). This contains important configuration information, regardless of your application server.
- Please be aware of BellaDati's [Supported platforms](#) page, which indicates what application servers are supported by BellaDati.
- Also be aware that we do not recommend deploying Multiple BellaDati Applications in a Single application server.

Installing BellaDati on Glassfish 2.1

 This guide describes how to install BellaDati WAR on [GlassFish 2.1 and above](#). You can download it from [GlassFish's download site](#). Please refer to [Sun GlassFish Enterprise Server 2.1 Installation Guide](#) when you want to get more informations about how to install GlassFish.

The BellaDati WAR is plain BellaDati application packaged in WAR archive. It must be deployed into an existing application server and needs to be connected to [data warehouse](#) running on existing database server. If you have BellaDati virtual appliance, follow the [Installing BellaDati virtual appliance](#) guide instead of the instructions below.

Before You Begin

Please read the following important notes before you begin installing BellaDati on GlassFish:


- Make sure you have the right version of JAVA. See [Installing JAVA](#) for more details.
- Deploying multiple BellaDati WAR application in a single GlassFish application server is not supported. We do not test this configuration and upgrading any of the applications (even for point releases) is likely to break it.


On this page:

- [Before You Begin](#)
- [1. Download and extract BellaDati WAR Archive](#)
- [2. Configure BellaDati WAR](#)
- [3. Update Your GlassFish Installation's Libraries for BellaDati](#)
- [4. Configure GlassFish](#)
 - [4.1 Java environment](#)
 - [4.2 Fix Memory settings in GlassFish](#)
 - [4.4 Configure the GlassFish's JDBC resource](#)
- [5. Deploying BellaDati into GlassFish](#)
 - [5.1 Using the `asadmin` command](#)
 - [5.2 Using Glassfish's Administration console](#)
- [6. Run the Setup Wizard](#)

1. Download and extract BellaDati WAR Archive

Download the BellaDati WAR distribution as TAR/GZ archive from [BellaDati download page](#) and extract its content using a tool such as 7-zip for Windows or Linux's/GNU tar tools. TAR/GZ archive contains single file - `belladati.war`.

 Avoid using Windows' built-in file extraction tool! This tool silently fails to extract files with long names. Other users have also reported problems with WinRAR.

 Avoid using Solaris' default tar utility! Please use GNU tar on this operating system to extract BellaDati, as GNU tar handles long filenames better.

2. Configure BellaDati WAR

You have successfully downloaded and extracted the BellaDati WAR distribution as described in the previous step. BellaDati WAR configuration

consists from editing a configuration file inside the WAR archive. To do it, please follow this easy steps:

1. Extract `application.properties` file from WAR using a tool such as 7-zip for Windows or Linux's `unzip` tool.

```
unzip belladati.war WEB-INF/classes/conf/application.properties
```

2. Edit extracted file `WEB-INF/classes/conf/application.properties`. The default configuration should look like:

```
# APPLICATION properties
application.url=http://localhost
application.production-mode=true
application.secure-mode=false

# JDBC properties
jdbc.resource=jdbc/belladati_db

# SMTP properties
smtp.server.address=localhost
smtp.server.username=
smtp.server.password=

# EMAIL properties
email.sender=support@belladati.com
email.footer=TRGIMAN s.r.o.

# OpenID properties
application.registration=http://www.belladati.com/try
application.registration.openid=

# Google analytics properties
application.google-site-verification=
application.google-analytics-account=
application.google-analytics-account.mercato=

# Lucene index dir
lucene.indexes=/tmp/indexes
```



Parameters are explained in [BellaDati WAR Configuration Overview](#).

3. Save configuration file `WEB-INF/classes/conf/application.properties` and update the WAR archive. Linux `unzip` command looks like:

```
zip -d belladati.war WEB-INF/classes/conf/application.properties
zip -u belladati.war WEB-INF/classes/conf/application.properties
```

3. Update Your GlassFish Installation's Libraries for BellaDati

Your GlassFish installation requires an appropriate JDBC driver to allow BellaDati to communicate with the database. To add this JDBC driver to GlassFish, refer to the appropriate instructions:

- [Copy the PostgreSQL JDBC Driver to GlassFish](#)

The driver's JAR file should be copied into the `$GLASSFISH_HOME/lib` directory.

4. Configure GlassFish

4.1 Java environment

Java environment is automatically setup during the GlassFish unbundle and configuration. You can find the current configuration in `$GLASSFISH_HOME/conf/asenv.conf` file. Entry `AS_JAVA` should point to directory, where the JDK is installed, for example:

```
AS_JAVA="/opt/jdk1.6.0_27/
```

4.2 Fix Memory settings in GlassFish

Memory settings need to be modified in GlassFish to avoid the following issue:

- BellaDati requires more memory than what GlassFish provides by default. This may lead to `OutOfMemory` errors when running BellaDati if these memory settings are not increased.

To prevent this issue, follow the instructions below.

If you are using the command line, execute the following commands:

```
$GLASSFISH_HOME/bin/asadmin create-jvm-options \ "-Xmx2048m\"  
$GLASSFISH_HOME/bin/asadmin create-jvm-options \ "-XX:MaxPermSize=512m\"
```

Or login into the Glassfish's Administration console (by default available at port 4848 with login admin/adminadmin) and open the Application server's JVM settings menu:



For other environments and more information on memory settings, see [System requirements](#).

4.4 Configure the GlassFish's JDBC resource



If you want to get more informations about the GlassFish's JDBC resources, visit the [Sun GlassFish Enterprise Server 2.1 Administration Guide](#)

BellaDati uses the JDBC resource, which handles the connection to the database subsystem. Although BellaDati supports own JDBC connecting to the database directly, we recommend to use GlassFish's JDBC resources to avoid performance problems and to have more control of the database transactions management. You can setup the JDBC connection pool and JDBC resource following these steps:

4.4.1 Using the `asadmin` command

1. Start GlassFish
2. Create JDBC connection pool:

```
$GLASSFISH_HOME/bin/asadmin create-jdbc-connection-pool \  
--datasourceclassname org.postgresql.ds.PGConnectionPoolDataSource \  
--restype javax.sql.ConnectionPoolDataSource --maxpoolsize 200 \  
--isconnectvalidatereq=true --validationmethod table --validationtable pg_tables \  
--property DatabaseName=belladati:User=belladati_dbuser:Password=password:ServerName=db_host  
belladati_jdbcpool
```

The `datasourceclassname` and `property` are vendor specific parameters. This example shows settings for PostgreSQL. Find the appropriate setting for your database:

- [PostgreSQL connection parameters for GlassFish](#)

3. Create JDBC resource:

```
$GLASSFISH_HOME/bin/asadmin create-jdbc-resource --connectionpoolid belladati_jdbcpool
jdbc/belladati_db
```

The jdbc/belladati_db corresponds with the parameter jdbc.resource in BellaDati's application.properties.

4.4.2 Using Glassfish's Administration console

You can also use the Glassfish's Administration console as described in the previous step:

1. Login to Administration console
2. Open the Resources->JDBC->Connection pools and click on New button.
3. Fill the name, type and database vendor fields and click next:



4. Fill the database connection parameters



Find the appropriate setting for your database:

- [PostgreSQL connection parameters for GlassFish](#)

Recommended connection parameters

Resource Type	javax.sql.ConnectionPoolDataSource
Database Vendor	depends on database vendor
Pool settings	
Initial and minimum pool size	8
Maximum pool size	200
Pool resize quantity	2
Idle timeout	300s
Max wait time	60000ms
Connection validation	
Connection validation	required
Validation method	table
Table name	depends on database vendor
On any failure	not checked
Non component callers	disabled

5. Deploying BellaDati into GlassFish

5.1 Using the `asadmin` command

In order to deploy BellaDati, execute this command:

```
$GLASSISH_HOME/bin/asadmin deploy --user admin /belladati.war
```

This command performs deploying BellaDati WAR on the `/belladati` context.

5.2 Using Glassfish's Administration console

You can use the Admin Console to deploy BellaDati into GlassFish.

1. Login to Administration console.
2. Open the Applications component.
3. Go to the page for the type of application or module.
4. Click on the Deploy button.
5. On this page, you type the path to the `belladati.war` file or the exploded directory structure.



More detailed description of the deployment process is described in [Sun GlassFish Enterprise Server 2.1 Application Deployment Guide](#).


6. Run the Setup Wizard



Point your browser to <http://localhost:8080/belladati>

You should now see the [Setup Wizard](#), which will take you through the BellaDati's setup procedure.

Installing BellaDati on Jetty 6

 This guide describes how to install BellaDati WAR on Jetty 6 and above.

The BellaDati WAR is plain BellaDati application packaged in WAR archive. It must be deployed into an existing application server and needs to be connected to [data warehouse](#) running on existing database server. If you have BellaDati virtual appliance, follow the [Installing BellaDati virtual appliance](#) guide instead of the instructions below.

Before You Begin

Please read the following important notes before you begin installing BellaDati on Jetty 6:


- Make sure you have the right version of JAVA. See [Installing JAVA](#) for more details.
- Deploying multiple BellaDati WAR application in a single Jetty container is not supported. We do not test this configuration and upgrading any of the applications (even for point releases) is likely to break it.
- Finally, we recommend not deploying any other applications in the same Jetty container that runs BellaDati, especially if these other applications have large memory requirements or require additional libraries in Jetty's lib subdirectory. There are also a number of practical reasons why we do not support deploying multiple applications in a single Jetty container. Firstly, you must shut down Jetty to upgrade any application and secondly, if one application crashes, the other applications running in that Jetty container will be inaccessible.


On this page:

- [Before You Begin](#)
- [1. Download and extract BellaDati WAR Archive](#)
- [2. Configure BellaDati WAR](#)
- [3. Update Your Jetty Installation's Libraries for BellaDati](#)
- [4. Configure Jetty](#)
 - [4.1 Java environment](#)
 - [4.2 Fix Memory settings in Jetty](#)
- [5. Start Jetty](#)
- [6. Run the Setup Wizard](#)

1. Download and extract BellaDati WAR Archive

Download the BellaDati WAR distribution as TAR/GZ archive from [BellaDati download page](#) and extract its content using a tool such as 7-zip for Windows or Linux's/GNU tar tools. TAR/GZ archive contains single file - `belladati.war`.

 Avoid using Windows' built-in file extraction tool! This tool silently fails to extract files with long names. Other users have also reported problems with WinRAR.

 Avoid using Solaris' default tar utility! Please use GNU tar on this operating system to extract BellaDati, as GNU tar handles long filenames better.

2. Configure BellaDati WAR

You have successfully downloaded and extracted the BellaDati WAR distribution as described in the previous step. BellaDati WAR configuration consists from editing a configuration file inside the WAR archive. To do it, please follow this easy steps:

1. Extract `application.properties` file from WAR using a tool such as 7-zip for Windows or Linux's unzip tool.

```
unzip belladati.war WEB-INF/classes/conf/application.properties
```

2. Edit extracted file `WEB-INF/classes/conf/application.properties`. The default configuration should look like:

```
# APPLICATION properties
application.url=http://192.168.1.100
application.production-mode=true
application.secure-mode=false

# JDBC properties
jdbc.driverClassName=org.postgresql.Driver
jdbc.url=jdbc:postgresql://db_host:5432/belladati_db
jdbc.username=belladati_dbuser
jdbc.password=password
jdbc.resource=jdbc/belladati_db

# SMTP properties
smtp.server.address=localhost
smtp.server.username=
smtp.server.password=

# EMAIL properties
email.sender=support@belladati.com
email.footer=TRGIMAN s.r.o.

# OpenID properties
application.registration=http://www.belladati.com/try
application.registration.openid=

# Google analytics properties
application.google-site-verification=
application.google-analytics-account=
application.google-analytics-account.mercato=

# Lucene index dir
lucene.indexes=/tmp/indexes
```

 Parameters are explained in [BellaDati WAR Configuration Overview](#).

3. Save configuration file `WEB-INF/classes/conf/application.properties` and update the WAR archive. Linux unzip command looks like:

```
zip -d belladati.war WEB-INF/classes/conf/application.properties
zip -u belladati.war WEB-INF/classes/conf/application.properties
```

4. Copy the `belladati.war` to `$JETTY_HOME/webapps`.

3. Update Your Jetty Installation's Libraries for BellaDati

Your Jetty installation requires an appropriate JDBC driver to allow BellaDati to communicate with the database. To add this JDBC driver to Jetty, refer to the appropriate instructions:

- [Copy the PostgreSQL JDBC Driver to Jetty](#)

The driver's JAR file should be copied into the `$JETTY_HOME/lib/ext` directory.

4. Configure Jetty

4.1 Java environment

Jetty looks for the JAVA environment searching the path specified in `JAVA_HOME` variable. It should be set and point to the JDK directory. Before you start Jetty, make sure you have configured this variable. On linux systems you can do it in this way:

```
$ export JAVA_HOME=/path/to/jdk
```

You can also add `JAVA_HOME` parameter to `/etc/default/jetty`. The `JAVA_HOME` will be set when you start Jetty.

4.2 Fix Memory settings in Jetty

Memory settings need to be modified in Jetty to avoid the following issue:

- BellaDati requires more memory than what Jetty provides by default. This may lead to `OutOfMemory` errors when running BellaDati if these memory settings are not increased.

To prevent this issue, follow the instructions below.

Set the `JAVA_OPTIONS` variable:

```
$ export JAVA_OPTIONS="-Xms1024m -Xmx2048m -XX:MaxPermSize=512m"
```

Or set the `JAVA_OPTIONS` parameter to `/etc/default/jetty` file:

```
JAVA_OPTIONS="-Xms1024m -Xmx2048m -XX:MaxPermSize=512m"
```

If you are using the Java Service Wrapper, follow these instructions:

Edit Jetty's `bin/jetty-service.conf` file (or create this file if it does not exist) and add the following to this file:

```
wrapper.java.additional.3=-Xms1024m  
wrapper.java.additional.4=-Xmx2048m  
wrapper.java.additional.5=-XX:MaxPermSize=512m  
wrapper.java.maxmemory=2048
```

For other environments and more information on memory settings, see [System requirements](#).



You can download the Java Service Wrapper from [this site](#).

5. Start Jetty

(Only required if Jetty is not running as a service.)

BellaDati should now be ready to run in Jetty. To start up BellaDati, start (or restart) the Jetty server with Jetty's `bin/jetty.sh run` or `bin/Jetty-Service.exe` scripts.

6. Run the Setup Wizard



Point your browser to <http://localhost:8080/belladati>

You should now see the [Setup Wizard](#), which will take you through the BellaDati's setup procedure.

Installing BellaDati on Tomcat 6



This guide describes how to install BellaDati WAR on [Tomcat 6.0](#). Tomcat container can be downloaded from the [Apache site](#).

The BellaDati WAR is plain BellaDati application packaged in WAR archive. It must be deployed into an existing application server and needs to be connected to [data warehouse](#) running on existing database server. If you have BellaDati virtual appliance, follow the [Installing BellaDati virtual appliance](#) guide instead of the instructions below.

Before You Begin

Please read the following important notes before you begin installing BellaDati on Tomcat 6.0:


- Make sure you have the right version of JAVA. See [Installing JAVA](#) for more details.
- Tomcat 6.0.24 contains a critical bug. Please use 6.0.32 instead.
- Deploying multiple BellaDati WAR application in a single Tomcat container is not supported. We do not test this configuration and upgrading any of the applications (even for point releases) is likely to break it.
- Finally, we recommend not deploying any other applications in the same Tomcat container that runs BellaDati, especially if these other applications have large memory requirements or require additional libraries in Tomcat's lib subdirectory. There are also a number of practical reasons why we do not support deploying multiple applications in a single Tomcat container. Firstly, you must shut down Tomcat to upgrade any application and secondly, if one application crashes, the other applications running in that Tomcat container will be inaccessible.


On this page:

- [Before You Begin](#)
- [1. Download and extract BellaDati WAR Archive](#)
- [2. Configure BellaDati WAR](#)
- [3. Update Your Tomcat Installation's Libraries for BellaDati](#)
- [4. Configure Tomcat](#)
 - [4.1 Set JAVA environment](#)
 - [4.2 Configure Tomcat's context](#)
 - [4.3 Set Tomcat home directory permissions](#)
 - [4.4 Fix Memory settings in Tomcat](#)
 - [4.5 Modify Tomcat's `server.xml`](#)
- [5. Start Tomcat](#)
- [6. Run the Setup Wizard](#)

1. Download and extract BellaDati WAR Archive

Download the BellaDati WAR distribution as TAR/GZ archive from [BellaDati download page](#) and extract its content using a tool such as 7-zip for Windows or Linux's/GNU tar tools. TAR/GZ archive contains single file - `belladati.war`.

 Avoid using Windows' built-in file extraction tool! This tool silently fails to extract files with long names. Other users have also reported problems with WinRAR.

 Avoid using Solaris' default tar utility! Please use GNU tar on this operating system to extract BellaDati, as GNU tar handles long filenames better.

2. Configure BellaDati WAR

You have successfully downloaded and extracted the BellaDati WAR distribution as described in the previous step. BellaDati WAR configuration consists of unpacking and editing a configuration file placed inside the WAR archive. To do it, please follow this easy steps:

1. Extract `application.properties` file from WAR using a tool such as 7-zip for Windows or Linux's unzip tool.

```
unzip belladati.war WEB-INF/classes/conf/application.properties
```

2. Edit extracted file `WEB-INF/classes/conf/application.properties`. The default configuration should look like:


```
# APPLICATION properties
application.url=http://localhost:8080
application.production-mode=true
application.secure-mode=false

# JDBC properties
jdbc.resource=java:/comp/env/jdbc/belladati_db

# SMTP properties
smtp.server.address=localhost
smtp.server.username=
smtp.server.password=

# EMAIL properties
email.sender=support@belladati.com
email.footer=TRGIMAN s.r.o.

# OpenID properties
application.registration=http://www.belladati.com/try
application.registration.openid=

# Google analytics properties
application.google-site-verification=
application.google-analytics-account=
application.google-analytics-account.mercato=

# Lucene index dir
lucene.indexes=/tmp/indexes
```



Parameters are explained in [BellaDati WAR Configuration Overview](#).

3. Save configuration file `WEB-INF/classes/conf/application.properties` and update the WAR archive. Linux unzip command looks like:

```
zip -d belladati.war WEB-INF/classes/conf/application.properties
zip -u belladati.war WEB-INF/classes/conf/application.properties
```

3. Update Your Tomcat Installation's Libraries for BellaDati

Your Tomcat installation requires an appropriate JDBC driver to allow BellaDati to communicate with the database. To add this JDBC driver to Tomcat, refer to the appropriate instructions:

- [Copy the PostgreSQL JDBC Driver to Tomcat](#)

4. Configure Tomcat

4.1 Set JAVA environment

Make sure the `$TOMCAT_HOME/conf/tomcat6.conf` file points to right version of **JAVA**:

```
JAVA_HOME="/opt/java64"
```

4.2 Configure Tomcat's context

A BellaDati 'context' now needs to be set up in Tomcat. To do this:

1. Create the directory structure `conf/Catalina/localhost/` within your Tomcat installation directory.
2. Create the `belladati.xml` file and copy it to `{{conf/Catalina/localhost}}` subdirectory of your Tomcat installation directory (created in the previous step).

```

<Context path="/belladati" docBase="path/to/belladati-2.x.war" debug="0" useHttpOnly="true">

<Resource name="jdbc/belladati_db" auth="Container" type="javax.sql.DataSource"
  driverClassName="org.postgresql.Driver"
  url="jdbc:postgresql://db_host:5432/belladati_db"
  username="belladati_dbuser" password="password"
  maxActive="20" maxIdle="10" maxWait="-1"/>

</Context>

```

If you are installing in Windows, make sure that the paths you specify for the location of the WAR file and database are full paths with drive letters (e.g. C:\path\to\belladati-2.x.war).

4.3 Set Tomcat home directory permissions

By default BellaDati is creating several directories within the Tomcat home folder. Tomcat should run under user, which has read/write access to this directory.

4.4 Fix Memory settings in Tomcat

Memory settings need to be modified in Tomcat to avoid the following issues:

- BellaDati requires more memory than what Tomcat provides by default. This may lead to `OutOfMemory` errors when running BelladDati if these memory settings are not increased.

To prevent these issues, follow the appropriate instructions for your operating system below.

For Linux/Solaris

Edit Tomcat's `bin/setenv.sh` file (or create this file if it does not exist) and add the following to this file:

```

export CATALINA_OPTS="$CATALINA_OPTS\ -Dmail.mime.decodeparameters=true -Xms1024m -Xmx2048m
-XX:MaxPermSize=512m"

```

The other way is to edit the `$TOMCAT_HOME/conf/tomcat6.conf` file:

```

JAVA_OPTS="-Dmail.mime.decodeparameters=true -Xms1024m -Xmx2048m -XX:MaxPermSize=512m"

```

For Windows

If Tomcat is not installed as a service:

- Edit Tomcat's `bin/setenv.bat` file (or create this file if it does not exist) and add the following to this file:

```

set CATALINA_OPTS=%CATALINA_OPTS% -Dmail.mime.decodeparameters=true -Xms1024m -Xmx2048m
-XX:MaxPermSize=512m\

```

If Tomcat is installed and running as a service:

1. Right-click Tomcat's system tray icon and select 'Configure' from the resulting popup menu, which opens the 'Apache Tomcat 6 Properties' dialog box.
2. In this dialog box, click on the 'Java' tab and specify the following values:

Field	Value
Java Options (append to the existing value)	-Dmail.mime.decodeparameters=true
Initial memory pool	1024

Maximum memory pool	2048
---------------------	------

For other environments and more information on memory settings, see [System requirements](#).

4.5 Modify Tomcat's `server.xml`


In order for BellaDati to correctly display internationalised characters in user and group names, you need to modify the `conf/server.xml` file in your Tomcat installation directory by specifying the `URIEncoding="UTF-8"` property within the connector definition for your HTTP protocol.

The connector definition is specified by the following element in your `server.xml` file:

```
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443"/>
```

You should modify this element by specifying the `URIEncoding="UTF-8"` attribute:

```
<Connector port="8080" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" URIEncoding="UTF-8"/>
```

 Please Note:


- Since this property must be specified at the connector level for your application server, this setting will effect all other web applications deployed to the same application server installation running BellaDati. While this setting should not adversely effect these other web applications, you should be aware of this point.
- BellaDati will run fine without this property set. However, you will run into issues if a user or group is created which contains international characters. Hence, it is recommended that you set this property.

5. Start Tomcat

(Only required if Tomcat is not running as a service.)

BellaDati should now be ready to run in Tomcat. To start up BellaDati, start (or restart) the Tomcat server with Tomcat's `bin/startup.sh` or `bin/startup.bat` scripts.

6. Run the Setup Wizard

 Point your browser to `http://localhost:8080/belladati`

You should now see the [Setup Wizard](#), which will take you through the BellaDati's setup procedure.

BellaDati WAR Configuration Overview

- [Parameters in `application.properties`](#)
- [Logging services](#)
- [LDAP Authentication](#)

Parameters in `application.properties`

File `application.properties` is located in BellaDati WAR archive. To get them, you can use tools such as 7-zip for Windows or Linux's/GNU unzip tools.

Parameter name	Description
<code>application.url</code>	Specifies the URL, on which is BellaDati running, default value is http://localhost
<code>application.production-mode</code>	Determines the production usage. The default value is <code>true</code> .

<code>application.secure-mode</code>	Determines whether BellaDati runs in SSL mode.
<code>jdbc.driverClassName</code>	The requested JDBC driver, which BellDati uses for database connections. Supported drivers are: <code>org.postgresql.Driver</code>
<code>jdbc.url</code>	If you don't have JNDI data source configured (supported for GlassFish 2.1 and Jetty 6 only), you can use the <code>jdbc.url</code> , <code>jdbc.username</code> and <code>jdbc.password</code> parameters. BellaDati creates JNDI datasource automatically. The URL should look like: <code>jdbc:postgresql://db_host:5432/belladati_db</code> Visit the Setup BellaDati data warehouse for more informations.
<code>jdbc.username</code>	Database user, which is used by BellDati to access the database. This parameter is used, when <code>jdbc.url</code> is set.
<code>jdbc.password</code>	Database user password, which is used by BellDati to access the database. This parameter is used, when <code>jdbc.url</code> is set.
<code>jdbc.resource</code>	The name of the JNDI resource, for example <code>jdbc/belladati_db</code>
<code>smtp.server.address</code>	URL of the SMTP server used for mailing. Default value is <code>localhost</code> .
<code>smtp.server.username</code>	Username for accessing the SMTP server.
<code>smtp.server.password</code>	Password for accessing the SMTP server.
<code>email.sender</code>	This parameter allows you to change the email sender address. <code>support@belladati.com</code> is used by default.
<code>email.footer</code>	Value of this parameter is automatically appended to all outgoing emails. The default value is <code>TRGIMAN S.R.O..</code>
<code>lucene.indexes</code>	Specifies the directory for indexes. Default value is <code>/tmp/indexes</code> .

Logging services

Logging system in BellaDati is provided by the [LogBack](#) framework. The location of the resulting log files, where the log messages are written, can be managed by changing the `WEB-INF/classes/logback.xml` configuration file contained in `belladati.war`.

BellaDati uses 3 appenders: `web`, `core` and `queries`. To change the log files storage path, you have to unzip the mentioned file and edit the `<file>` tag. For example, the configuration can look like:

```

<appender name="LOG-FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>../logs/belladati/belladati_web.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <fileNamePattern>../logs/belladati/belladati_web.%d{yyyy-MM-dd}.log</fileNamePattern>
  </rollingPolicy>
  <encoder>
    <pattern>%d{HH:mm:ss,SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
  </encoder>
</appender>

<appender name="CORE-LOG-FILE" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>../logs/belladati/core/ds-core.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <fileNamePattern>../logs/belladati/core/ds-core.%d{yyyy-MM-dd}.log</fileNamePattern>
  </rollingPolicy>
  <encoder>
    <pattern>%d{HH:mm:ss,SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
  </encoder>
</appender>
<appender name="CORE-LOG-FILE-QUERIES" class="ch.qos.logback.core.rolling.RollingFileAppender">
  <file>../logs/belladati/core/ds-failed-queries.log</file>
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
    <fileNamePattern>logFile.%d{yyyy-MM-dd}.log</fileNamePattern>
  </rollingPolicy>
  <encoder>
    <pattern>%d{HH:mm:ss,SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
  </encoder>
</appender>

```

If you experience some problems using BellaDati or something doesn't work like you are expecting, these files should be appended to the issue you can create at <http://support.trgiman.eu>.

LDAP Authentication

BellaDati supports customizable authentications providers. On of the basic providers is the Active Directory and LDAP authentication provider. To enable this authentication mode, follow instructions below:

1. Unzip the WEB-INF/classes/conf/beans-common.xml configuration file from
2. Replace the element with id="userAuthenticityProvider with this content:

```

<bean id="userAuthenticityVerifier" class="cz.trgiman.belladati.auth.ADUserAuthenticityVerifier"
autowire="byType">
  <property name="createNotExistingAccounts" value="true" />
  <property name="workspaceId" value="1" />
  <property name="domains">
    <map>
      <entry>
        <key><value>domain1</value></key>
        <bean class="cz.trgiman.belladati.auth.DomainSpecification">
          <property name="url" value="LDAP://host1:389/dc=xyz" />
          <property name="bindDN" value="bindUser" />
          <property name="bindPassword" value="bindUserPassword" />
          <property name="loginAttribute" value="sAMAccountName" />
        </bean>
      </entry>
      <entry>
        <key><value>domain2</value></key>
        <bean class="cz.trgiman.belladati.auth.DomainSpecification">
          <property name="url" value="LDAP://host2:389/dc=xyz" />
          <property name="bindDN" value="bindUser2" />
          <property name="bindPassword" value="bindUserPassword2" />
          <property name="loginAttribute" value="cn" />
        </bean>
      </entry>
    </map>
  </property>
</bean>

```

3. Add following parameter to `WEB-INF/classes/conf/application.properties` file:

```
application.login.domains=domain1,domain2
```

4. Update the `belladati.war` archive.

Installing BellaDati virtual appliance

BellaDati virtual appliance is available in the following formats:

- VMDK - stands for VMWare's Virtual machine disk format
- OVF - the Open Virtualisation Format

Both have BellaDati configured and running on PostgreSQL, as well as frontend available.

BellaDati virtual appliance has been built using the [SUSE Studio](#).

The pages listed below contain information on installing BellaDati as an virtual appliance:

- [Installing BellaDati on VMWare ESX and ESXi](#)
- [Installing BellaDati on VMWare Player](#)
- [Launching BellaDati virtual appliance](#)

How does VMDK compare to OVF?

VMDK is a file format that only encodes a single virtual disk from a virtual machine. A VMDK does not contain information about the virtual hardware of a machine, such as the CPU, memory, disk, and network information. A virtual machine may include multiple virtual disks or VMDKs. An administrator who wishes to deploy a virtual disk must then configure all of this information, often manually, using incomplete documentation.

The OVF format, on the other hand, provides a complete specification of the virtual machine. This includes the full list of required virtual disks plus the required virtual hardware configuration, including CPU, memory, networking, and storage. An administrator can quickly provision this virtual machine into virtual infrastructure with little or no manual intervention. In addition, the OVF is a standards-based, portable format that allows the user to deploy this virtual machine in any hypervisor that supports OVF.

Installing BellaDati on VMWare ESX and ESXi



This guide is intended for installing BellaDati on [VMware vSphere Hypervisor ESX/ESXi](#) version 4.0 and above using the vSphere Client. For more info on VMWare products, please visit www.vmware.com.

The BellaDati virtual appliance is BellaDati preconfigured with a copy of GlassFish application server and PostgreSQL database. If you have BellaDati WAR, follow the [Installing BellaDati WAR](#) guide instead of the instructions below.

On this page:

- [System requirements](#)
- [Downloading BellaDati OVF](#)
- [Deploying OVF image](#)
- [Next steps](#)

System requirements

BellaDati requires both physical and disk memory. 4094 MB of physical memory and 5GB of free disk space could be a good starting point. However, the amount of required disk memory obviously depends on proposed system load. Amount of memory allocated for the connection depends on configuration of the database engine.

The more physical memory you have, the faster the BellaDati works!



Find out accurate system requirements for your purpose
Visit the [system requirements](#) section.

Standard configuration overview

Depending on the proposed system load there are several OVF configurations ready for download.

	Basic configuration 4/20GB	Medium configuration 8/100GB
OS	openSUSE 11.4 64-bit x86	openSUSE 11.4 64-bit x86
CPU cores	1	1
Memory	4096MB	8192MB
Disk capacity	20GB	100GB
Volume manager	LVM	LVM



Default system configuration is proposed for medium loaded system
For advanced and high loaded system usage (hundreds of users are analysing millions records) configuration change may be required.

Downloading BellaDati OVF

In this chapter you will learn how to install BellaDati on-premise version on a single server using an OVF image. Remember that you need to be registered into my.trgiman.eu to get the OVF image and your licence key. For downloading the OVF distribution, please visit the [download](#) section.

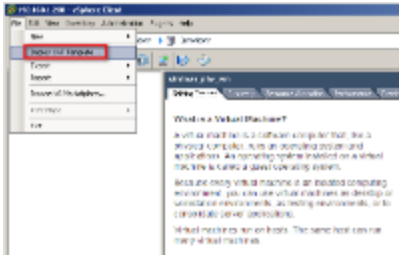


The downloaded file is in TAR/GZip format, so before you start to deploy it into ESX/ESXi server, you have to extract it from this archive.

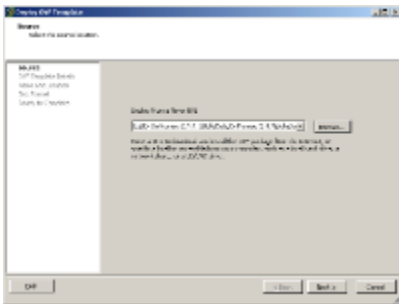
Deploying OVF image

Moreover you need to obtain VMWare vSphere Client to deploy the OVF image to ESX or ESXi environment (4.0+ version is required). Additionally you will need one free IPv4 for network configuration. Let presume that you have downloaded the OVF image and successfully installed the vSphere Client.

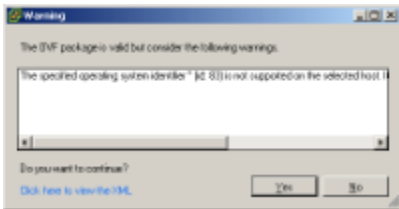
1. Now open the VMWare vSphere Client and click on the File - Deploy OVF Template.



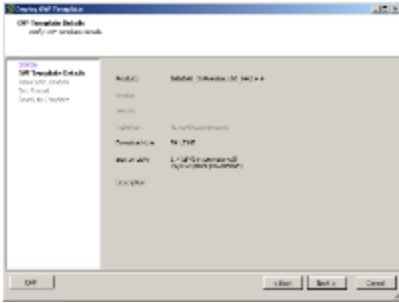
2. Then you need to specify source of the OVF image. Click on the Browse, find your downloaded and extracted OVF image and click on the next button.



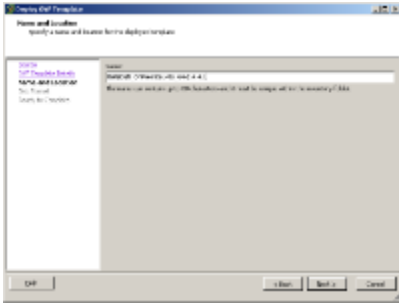
3. After the image is loaded you will see a short message containing warning - it's caused by some specific features of the OVF image which are not supported by VMWare. This features are not required for running BellaDati and you can ignore the message by clicking on "Yes" button.



4. In next step just check the OVF image details and click on the "Next" button.

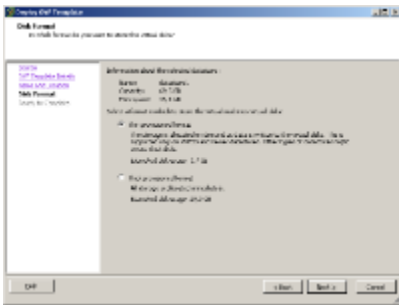


5. In this step please provide the desired name for the deployed OVF image and click on the "Next" button.

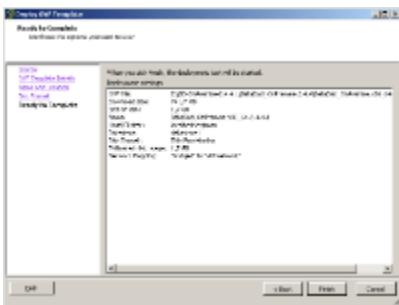


Note: When your server consists of more than one disk, please select also the desired datastore in this step.

6. Now you have to choose format for storing virtual discs. We recommend you to use "Thick provisioned format". Then click on the "Next" button.



7. Finally check all the deployment settings (it should be similar like on the picture below) and click on the "Finish" button.



8. You will be informed about finishing of deployment. At this moment BellaData OVF image is installed in your ESX/ESXi server.

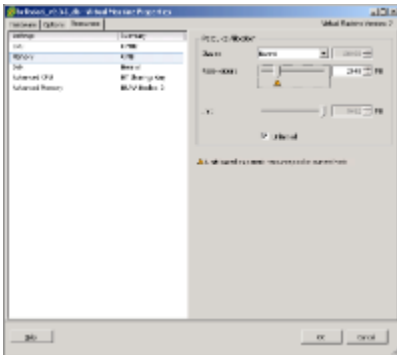


9. Optionally you can setup the resources reservation settings (Resources tab of the same window).



Reserve enough physical memory

It is recommended to set "Reservation" to value of your RAM memory space depending on selected image configuration (4096 or 8192MB). If you don't reserve these resources, lack of RAM memory can lead to system instability and may cause serious slowing down of application performance.



Next steps

Now your BellaDati OVF image installation is complete. Click on its name in the left tree structure and then click on Power On button ("play" symbol in the upper menu). BellaDati application will be started. It can take a few minutes.

Then follow the instructions at [Launching BellaDati virtual appliance](#).

Installing BellaDati on VMWare Player



This guide describes how to install BellaDati on on [VMWare Player](#) version 3.1 and above. It can be downloaded from www.vmware.com.

The BellaDati virtual appliance is BellaDati preconfigured with a copy of GlassFish application server and PostgreSQL database. If you have BellaDati WAR, follow the [Installing BellaDati WAR](#) guide instead of the instructions below.



Running BellaDati on VMWare Player is not recommended for production usage. This kind of installation is intended for testing purposes only.

On this page:

- [System requirements](#)
- [Downloading BellaDati VMDK](#)
- [Running BellaDati on VMWare Player](#)
- [Next steps](#)

System requirements

BellaDati requires both physical and disk memory. 4096 MB of physical memory and 5GB of free disk space could be a good starting point. However, the amount of required disk memory obviously depends on proposed system load. Amount of memory allocated for the connection depends on configuration of the database engine.

The more physical memory you have, the faster the BellaDati works!



Find out accurate system requirements for your purpose
Visit the [system requirements](#) section.

Standard configuration overview

Depending on the proposed system load there are several OVF configurations ready for download.

	Basic configuration 4/20GB	Medium configuration 8/100GB
OS	openSUSE 11.4 64-bit x86	openSUSE 11.4 64-bit x86
CPU cores	1	1
Memory	4096MB	8192MB
Disk capacity	20GB	100GB
Volume manager	LVM	LVM



Default system configuration is proposed for medium loaded system
For advanced and high loaded system usage (hundreds of users are analysing millions records) configuration change may be required.

Downloading BellaDati VMDK

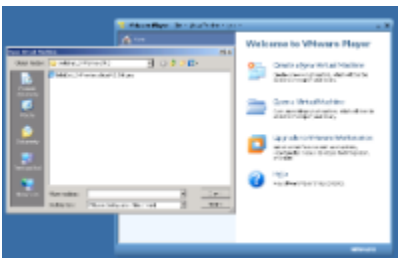
In this chapter you will learn how to install BellaDati on-premise version on a single server using an VMDK image. Remember that you need to be registered into my.trgiman.eu to get the VMDK image and your licence key. For downloading the VMDK distribution, please visit the download section.



The downloaded file is in TAR/GZip format, so before you run it on VMWare player, you have to extract it from this archive.

Running BellaDati on VMWare Player

Start VMWare Player and click on "Open a virtual machine". Locate the VMDK image you have downloaded from my.trgiman.eu and confirm the dialog.



After loading the VMDK image into VMWare player, you can start it clicking on "Play virtual machine". It can take a few minutes.



Next steps

Now your BellaDati VMDK installation is complete. Please follow the instructions at [Launching BellaDati virtual appliance](#).

Launching BellaDati virtual appliance

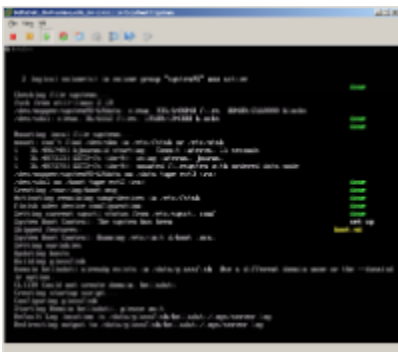
i This part describes launching of BellaDati virtual appliance for the first time after being installed into virtual environment as described in [Installing BellaDati on VMWare ESX and ESXi](#) and [Installing BellaDati on VMWare Player](#)

On this page:

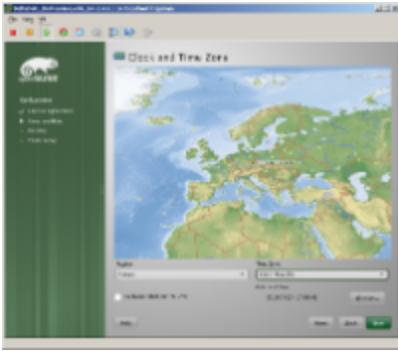
- [Launching BellaDati virtual appliance](#)
- [Network and other settings](#)
 - [Network configuration changing](#)
 - [Using Yast](#)
 - [Using CLI](#)
 - [Other settings](#)

Launching BellaDati virtual appliance

Once your BellaDati virtual appliance installation is complete, you can launch it in your virtual environment. Finally you will see a result of system startup in console. It should look like on the picture below:



Then you just need to confirm a licence agreement and choose appropriate time zone from the list or simply by clicking on the map.



After confirming you just need to wait a while for saving the system configuration. Then you will be able to start [BellaDati setup wizard](#) by entering your server IP address into your browser.



Network settings

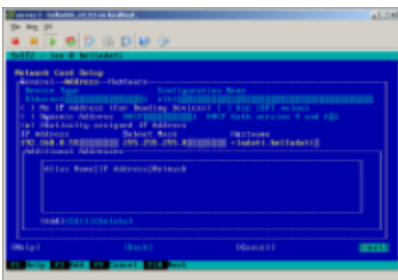
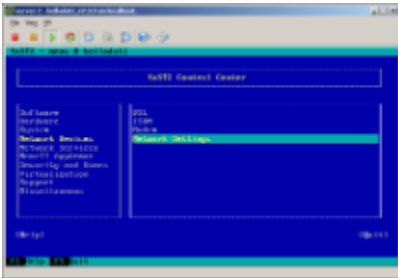
The default IP address is 192.168.1.100. When you want use BellaDati across your network, do not forget to configure the gateway and name servers. Please find more details in "Network and other settings" section.

Network and other settings

Network configuration changing

Using Yast

1. Open console window of launched instance or login remotely using SSH
2. Login as `root` with `belladati` password
3. Start Yast using the `yast` command
4. Enter the `Network Devices -> Network settings` menu
5. Change the IP configuration by editing the adapter settings (CTRL+i)
6. Modify the DNS settings (CTRL+s)
7. Modify the gateway settings (CTRL+u)



Using CLI

You need to adjust network settings before launching of BellaDati application (in our example we choose IP address 192.168.1.40/24 for demonstrational purposes)

1. Open console window of launched instance

2. Login as a "root" with "belladati" password
3. Adjust network interface - usually is available eth0 (you may easily use Yast application to avoid steps 3-5)

```
belladati:\~ # ifconfig eth0 192.168.1.40 net-mask 255.255.255.0 up
```

IP address and mask may be also set depending to specific needs of your infrastructure. Note: Availability of all interfaces in a single instance can be enabled by command:

```
belladati:\~ # ifconfig -a
```

4. Set default gateway (if it is necessary):

```
belladati:\~ # route add default gw 192.168.1.138
```

5. You need to change configuration also in /etc/sysconfig/network/ifcfg-eth0 file:

```
BOOTPROTO=static  
BROADCAST=192.168.1.255  
IPADDR=192.168.1.40/24  
STARTMODE=onboot  
USERCONTROL=no  
FIREWALL=yes
```

6. You check configuration by command:

```
belladati:\~ # ifconfig eth0  
eth0 Link encap:Ethernet HWaddr 00:0C:29:E1:70:91  
inet addr:192.168.1.40 Bcast:192.168.1.255 Mask:255.255.255.0  
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
RX packets:137 errors:0 dropped:0 overruns:0 frame:0  
TX packets:56 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:13358 (13.0 Kb) TX bytes:6798 (6.6 Kb)
```

7. Restart server using reboot command

Other settings

Operating system has firewall with following open ports: 22 (ssh), 80 (HTTP), 443 (HTTPS) and 4848 (Glassfish)

There are 3 logical partitions:

- /data - contains all the data of application server, BellaDati data-warehouse and indexes folders
- / - contains OS files, GlassFish installation and PostgreSQL and users folders (/home)
- swap - swap partition

JVM monitoring is available through JMX console.

Application logs are located in /data/glassfish/belladati/logs/belladati/ folder.

Setup BellaDati data warehouse

BellaDati requires a relational database to build its data warehouse.

The following instructions for configuring a connection to a BellaDati database apply to BellaDati WAR distributions:

- [Setup BellaDati with MySQL](#)

- [Setup BellaDati with Oracle](#)
- [Setup BellaDati with PostgreSQL](#)

Setup BellaDati with PostgreSQL

 These instructions will help you connect BellaDati to a [PostgreSQL 8.4+](#) database. It is intended for BellaDati WAR installations.

Before You Begin


Are You Migrating BellaDati to Another Server?

If you are migrating BellaDati to another server, create an export of your data as an XML backup. You will then be able to transfer data from your old database to your new database.

On this page:

- [Before You Begin](#)
 - [Are You Migrating BellaDati to Another Server?](#)
- [Create and Configure the PostgreSQL Database](#)
- [Copy the PostgreSQL JDBC Driver to Your Application Server](#)
- [PostgreSQL configuration](#)
- [PostgreSQL connection parameters for GlassFish](#)
- [Next steps](#)

Create and Configure the PostgreSQL Database

1. Create a database user which BellaDati will connect as (e.g. `belladati_dbuser`).  Remember this database user name, as it will be used to configure BellaDati's connection to this database in subsequent steps. Do not forget to configure the `pg_hba.conf` file, which controls the database access.

```
$ createuser -SDRP belladati_dbuser
```

2. Create a database for BellaDati to be used as data warehouse (e.g. `belladati_db`) with Unicode collation.

```
CREATE DATABASE belladati_db WITH ENCODING 'UNICODE';
```

Or from the command-line:

```
$ createdb -E UNICODE -O belladati_dbuser belladati_db
```

3. Ensure that the user has permissions to connect to the database, and to create and write to tables in the database.
4. Check if your database has the `plpgsql` language, if not, you can define it using this command:

```
CREATE LANGUAGE plpgsql
```

Or from the command-line:

```
$ createlang -U belladati_dbuser plpgsql belladati_db
```



When using default PostgreSQL configuration file `pg_hba.conf`, you have to execute this command by connecting via TCP socket:

```
createlang -h 127.0.0.1 -U belladati_dbuser plpgsql belladati_db
```



Remember this database name, as it will be used to configure BellaDati's connection to this database in subsequent steps.

Copy the PostgreSQL JDBC Driver to Your Application Server



Skip this step if you are using [BellaDati virtual appliance](#) distribution. BellaDati virtual appliance includes the PostgreSQL JDBC driver.

1. Download the PostgreSQL JDBC driver from <http://jdbc.postgresql.org/download.html>. (Obtain version 9.0 of the JDBC 4 driver.)
2. Add the PostgreSQL JDBC driver jar to the `lib/` directory of your application server.

PostgreSQL configuration

We are strongly recommending to perform the following PostgreSQL configuration changes in `$POSTGRES_HOME/data/postgresql.conf`. This example specifies parameters when you are running BellaDati and database on a single server with 4GB memory. 2GB of the are allocated for BellaDati application server running on JAVA (the `-Xmx` parameter). The rest should be available for operating system and database server.

Name	Value	Note
<code>max_connections</code>	50	
<code>shared_buffers</code>	512MB	min 1/4 RAM
<code>temp_buffers</code>	2MB	per session
<code>work_mem</code>	10MB	per client connection
<code>maintenance_work_mem</code>	64MB	
<code>max_stack_depth</code>	2MB	
<code>wal_buffers</code>	512kB	
<code>effective_cache_size</code>	512MB	
<code>log_statement</code>	none	



We are recommending set the `archive_mode` parameter to `off` and using of user defined `pg_dump` instead.

PostgreSQL connection parameters for GlassFish

User	The database user, which BellaDati uses for establishing the database connection
Password	Database user password
ServerName	Server name or IP address of the machine, where the database is running
DatabaseName	The database name



Do not forget to configure your PostgreSQL database instance's `pg_hba.conf` and `postgresql.conf` files for allowing incoming connections for desired user and database name.

Next steps

Continue configuring BellaDati WAR for your application server as described on:

- Installing BellaDati on Glassfish 2.1
- Installing BellaDati on Tomcat 6
- Installing BellaDati on Jetty 6

Running the setup wizard

You can setup BellaDati on-premise in the following easy steps.

1. Step: Entering setup wizard

After you have installed BellaDati WAR or BellaDati virtual appliance and have accessed its URL (eg. <http://localhost:8080> or <http://localhost:8080/belladati>) for the first time, you will be presented with a brief Setup Wizard to setup BellaDati.



If you are using the BellaDati Virtual Appliance, the default IP address is <http://192.168.1.100>.

BellaDati setup wizard will guide you through the setup process in a few simple steps.



On this page:

- 1. Step: Entering setup wizard
- 2. Step: License key
- 3. Step: Domain details
- 4. Step: Administrator Account
- 5. Step: First login

2. Step: License key

You are required to enter a valid license key before you can use BellaDati. You can obtain an 30-days evaluation license key which will allow BellaDati to run unrestricted for 30 days. To use your existing license key or obtain a new license key, follow the steps below:

- If you are a new user, you will need to create an account to generate an evaluation license key. Otherwise, log in to the my.belladati.com portal and navigate to the 'Licenses and tariffs' page.
- Once you have created an account or logged in with an existing account, the 'Licenses and tariffs' page will display with a list of your existing BellaDati licenses. If you do not have any BellaDati licenses, generate one by clicking on the 'New license'. A new evaluation license will be generated and displayed on the page.



- Copy the license key to your clipboard and paste it into the 'License Key' field of the Setup Wizard. After purchasing of a full version you will need to upgrade your licence.



Insert your licence key into prepared field of the wizard using CTRL+V and confirm by pressing "Continue" button.



3. Step: Domain details

In this step you need to specify your domain details. Domain name represents the name of your company or team - it will be displayed in upper right corner of application. Then select a language that will be used as a default language for users.



4. Step: Administrator Account

After completing first steps, the this page of the wizard sets up an administrator account. Email address will be used as your login name.



After confirming of your administration account, the setup process will be finished. Now you can visit documentation including basic and advanced tutorial movies, BellaDati news or directly move to the application login screen using according icons.




5. Step: First login

Now your BellaDati on-premise is ready. On the login screen just enter your login informations (use email address and password provided in the 4th step). After logging in you will see the basic desktop screen and welcome dashlet containing links to tutorials, news and BellaDati community sites again.



You can display information about your domain by clicking on your domain name in the upper right corner. Now your application is ready to use. If you want to know how to create new user accounts, change their access rights or how to import data, please watch our [video-tutorials](#).

Upgrading BellaDati

 This document describes the standard, recommended procedure for upgrading BellaDati to newer version.

If any of the following points applies to your situation, use the migration procedure to upgrade BellaDati instead:

- You are changing the location of your index and/or attachments path
- You are changing the operating system
- You are changing the database system

The pages listed below contain information on upgrading BellaDati:

- [Upgrading BellaDati virtual appliance](#)
- [Upgrading BellaDati WAR](#)

Upgrading BellaDati virtual appliance

Upgrading BellaDati installed as virtual appliance can be made using build-in application server administration console or using the linux shell. In both cases BellaDati WAR distribution is required.

On this page:

- [Upgrading using the administration console](#)

Upgrading using the administration console

Administration console is available as web application running on BellaDati host on port 4848.

Configure BellaDati WAR

The downloaded WAR archive contains `application.properties` with default settings. Before you start upgrading, you have to modify this configuration. To do it, please follow these easy steps:

1. Extract `application.properties` file from WAR using a tool such as 7-zip for Windows or Linux's `unzip` tool.

```
unzip belladati.war WEB-INF/classes/conf/application.properties
```

2. Edit extracted file `WEB-INF/classes/conf/application.properties`. The default configuration should look like:

```
# APPLICATION properties
application.url=http://localhost
application.production-mode=true
application.secure-mode=false

# JDBC properties
jdbc.resource=jdbc/belladati_db

# SMTP properties
smtp.server.address=localhost
smtp.server.username=
smtp.server.password=

# EMAIL properties
email.sender=support@belladati.com
email.footer=TRGIMAN s.r.o.

# OpenID properties
application.registration=http://www.belladati.com/try
application.registration.openid=

# Google analytics properties
application.google-site-verification=
application.google-analytics-account=
application.google-analytics-account.mercato=

# Lucene index dir
lucene.indexes=/tmp/indexes
```



Parameters are explained in [BellaDati WAR Configuration Overview](#).

3. Save configuration file `WEB-INF/classes/conf/application.properties` and update the WAR archive. Linux `unzip` command looks like:

```
zip -d belladati.war WEB-INF/classes/conf/application.properties
zip -u belladati.war WEB-INF/classes/conf/application.properties
```

Deploying BellaDati WAR

1. Login as `admin/adminadmin` into administration console (default address is http://belladati_host:4848).



2. In the left side menu choose `Applications->Web applications`. After clicking on `Web applications`, `BellaDati` should be contained in the list



3. Click on redeploy link, choose your BellaDati WAR file and proceed.



After these steps are completed, BellaDati upgrade process is finished.

Upgrading BellaDati WAR



This document describes the procedure for upgrading BellaDati running on Tomcat, GlassFish or Jetty application servers. If you are looking for procedure how to install BellaDati on these servers follow the instructions on [Installing BellaDati WAR](#) page.

Configure BellaDati WAR

The downloaded WAR archive contains `application.properties` with default settings. Before you start upgrading, you have to modify this configuration To do it, please follow this easy steps:

1. Extract `application.properties` file from WAR using a tool such as 7-zip for Windows or Linux's unzip tool.

```
unzip belladati.war WEB-INF/classes/conf/application.properties
```

2. Edit extracted file `WEB-INF/classes/conf/application.properties`. The default configuration should look like:

```
# APPLICATION properties
application.url=http://localhost
application.production-mode=true
application.secure-mode=false

# JDBC properties
jdbc.resource=jdbc/belladati_db

# SMTP properties
smtp.server.address=localhost
smtp.server.username=
smtp.server.password=

# EMAIL properties
email.sender=support@belladati.com
email.footer=BellaDati

# OpenID properties
application.registration=http://www.belladati.com/try
application.registration.openid=

# Google analytics properties
application.google-site-verification=
application.google-analytics-account=
application.google-analytics-account.mercato=

# Lucene index dir
lucene.indexes=/tmp/indexes
```



Parameters are explained in [BellaDati WAR Configuration Overview](#).

3. Save configuration file `WEB-INF/classes/conf/application.properties` and update the WAR archive. Linux unzip command looks like:

```
zip -d belladati.war WEB-INF/classes/conf/application.properties
zip -u belladati.war WEB-INF/classes/conf/application.properties
```

Deploying BellaDati WAR

Deploying new `belladati.war` doesn't need anything special, just copying the WAR archive into appropriate directory.

Tomcat and Jetty users have to stop the running server instance and replace the old `belladati.war` file with the new one.

GlassFish users have several options how to deploy the new `belladati.war`. The simplest way is to use the administration console - this process is briefly described in [Upgrading BellaDati virtual appliance](#).