

	<p>VO-Dance Installation Guide</p>	<p>Document No.</p>	<p>VO-Dance-001</p>
<p>Issue/Rev. No.</p>		<p>0.3-1</p>	
<p>Date</p>		<p>December 2013</p>	
<p>Page</p>		<p>1 of 9</p>	

VO-Dance Installation Guide

Document No.: VO-Dance-001

Issue/Rev. No.: 0.3-1

Date: 2013-12

Prepared by: Marco Molinaro

Author(s): Marco Molinaro, Francesco Cepparo

Approved by:

This document contains colour images, which can be printed in monochrome.

	VO-Dance Installation Guide	Document No. Issue/Rev. No. Date Page	VO-Dance-001 0.3-1 December 2013 2 of 9
---	-----------------------------	--	--

Document Approval

Name	Organization	Signature	Date

Document Status Sheet

1. Document Title: VO-Dance Installation Guide			
2. Issue	3. Revision	4. Date	5. Reason for change
0	1a	2011-06	M.Molinaro - initial version
0	2	2013-08	M.Molinaro - cleanup and improve (tests and feedback from F. Cepparo)
0	3	2013-11	M.Molinaro - cleanup after update to Glassfish v3, less code snippets and no figures. General installation notes.
0	3-1	2013-12	M.Molinaro – minor details from a new instance on ia2-vo server

	VO-Dance Installation Guide	Document No. Issue/Rev. No. Date Page	VO-Dance-001 0.3-1 December 2013 3 of 9
---	------------------------------------	--	--

Table of Contents

Document No.:	VO-Dance-001	1
Issue/Rev. No.:	0.3-1	1
Date:	2013-12	1
Prepared by:	Marco Molinaro	1
Author(s):	Marco Molinaro, Francesco Cepparo.....	1
Approved by:	1
This document contains colour images, which can be printed in monochrome.		1
Document Approval.....		2
Document Status Sheet		2
Table of Contents		3
1. Introduction.....		4
2. System Requirements.....		4
2.1.	Basic requirements	4
2.2.	VO-Dance distribution files	5
2.3.	Software Environment	5
3. Installation Details		5
3.1.	Initial Internal Database Setup	5
3.2.	VO-Dance Administration Interface Setup.....	6
3.3.	Glassfish Configuration and VO-Dance Deployment	7
4. Testing.....		8
Appendix.....		8
A.	Web Containers and Servers Port Numbers.....	8
B.	Apache HTTPD Configuration Notes.....	8
A.	Environment Versions.....	9

	VO-Dance Installation Guide	Document No.	VO-Dance-001
Issue/Rev. No.		0.3-1	
Date		December 2013	
Page		4 of 9	

1. Introduction

VO-Dance is a web application developed at INAF-OATs IA2 to allow easy publishing of VO related resources. This guide describes the requirements to install and run a VO-Dance instance starting from a tar gzip-ed distribution package including:

- the .war (Web Archives Repository) packaged web application (VO-Dance);
- the Django Python package to deploy the VO-Dance Administration Interface (VO-DAI);
- the MySQL script for the initial environment setup for VO-Dance to work (VO-DIDB).

For information on how to use VO-Dance to publish your services you will need the VO-Dance user manual (not available yet). However the distribution package builds a clean VO-Dance publishing engine that includes two working VO services' samples (1 SIA and 1 SCS service).

Sec. 2 provides an overview on system requirements while Sec. 3 guides the installer through the process of setting up the three tokens needed to have a working installation of the VO-Dance service publisher. Sec. 4 is a, as short as possible, testing solution to verify the installation. Appendices give some hint about possible conflicts in server ports (App. A) and information on the Apache configurations one may run upon (App. B).

2. System Requirements

For a VO-Dance installation you basically need a machine with enough resources to run a web application server, an HTTP server and a SQL server. This guide leads through a suggested installation based upon a Linux server (CentOS 6+, but has been tested also on Fedora 19) with a selected set of software distributions which contain all the needed tokens for VO-Dance.

This selection, however, is not mandatory and you can take advantage of software and resources already available on your machine (in that case custom changes may be needed). Moreover the VO-Dance instance can be spread on more than one single machine and can, at least in principle, be installed also on a Windows machine.

2.1. Basic requirements

The resources needed for VO-Dance (and provided through this installation guide) are:

- java JDK;
- an HTTP server (usually Apache);
- a Glassfish web application server (v.3, ask if you need a .war for v.2);
- a MySQL database (5.5 at least suggested);
- Python (v2);
- the Django python module;
- the MySQL JDBC (Java DataBase Connectivity) driver (version depends on MySQL server choice).

	VO-Dance Installation Guide	Document No. Issue/Rev. No. Date Page	VO-Dance-001 0.3-1 December 2013 5 of 9
---	------------------------------------	--	--

Appendix C contains some more detailed information on the supported version and related issues.

2.2. VO-Dance distribution files

The VO-Dance tar gzipped distribution package provided by IA2 contains:

- VO-Dance web application .war package;
- VO-Dance admin interface Django/Python package files (including WSGI and Apache config file skeletons) in a tar.gz package;
- VO-Dance startup internal DB including details and data for two working service examples (.sql dump file).

2.3. Software Environment

Oracle JDK (1.7 or 1.6 at least) has to be available in the server for Glassfish to set up and run properly. Glassfish v.3 has to be up and running when reaching web container configuration steps. A working MySQL server must be up and running, both for deploying the initial internal database schemas and for testing the connection between from the Glassfish web container and the VO-DAI to the SQL server. The MySQL JDBC connector (Java jar package) is needed for Glassfish to connect to the SQL server.

Python environment and Django framework are needed for the VO-DAI only. The MySQLdb (python-mysql) python module is required for VO-DAI to connect to the SQL server. The Apache mod_wsgi module is required for the Django based VO-DAI to be served by the httpd server.

A software stack, named DjangoStack, can solve many problems on a fresh server installation because it includes MySQL, Apache, Python and Django in an all-in-one installation process. It's up to the user to choose this solution or install single tokens.

Note: we suggest to create a new user on the server to run all the installation process (named *vodance*). This will help in managing the Glassfish web application container and keep the VO-Dance installation clean. However some configurations for the Apache web server may require root privileges.

3. Installation Details

Installation for the Java JDK, the Apache web server and its required modules, the Python language with Django module and dependencies, the MySQL DBMS and the Glassfish web container are not described here since they can change from server to server, depending on many details (OS flavour, package managers, distributions, versions, ...).

What you will find here are only notes and details specific to VO-Dance with some tips that may be useful in the process.

3.1. Initial Internal Database Setup

A SQL script is provided with the VO-Dance installation package. If you run it, e.g.

```
$ mysql -u root -p < VODance_MySQL_DB_setup.sql
```

	VO-Dance Installation Guide	Document No.	VO-Dance-001
		Issue/Rev. No.	0.3-1
		Date	December 2013
		Page	6 of 9

you'll end up with a couple of database schemas: *vodance* and *vodance_samples*. The first one is the VO-DIDB while the second one contains some sample data that allow the two sample services contained in the startup internal schema to work properly.

For MySQL the only other step to perform is the creation of a proper MySQL user granted with ALL PRIVILEGES on the two schemas above mentioned. This can be done, e.g form the standard command line MySQL client issuing commands similar to these ones:

```
mysql> create user vodance@localhost identified by 'a_password';
```

```
mysql> grant all privileges on vodance.* to vodance@localhost;
```

```
mysql> grant all privileges on vodance_samples.* to vodance@localhost;
```

The schema names are fixed by the SQL script used to initialize the internal DB. The MySQL user ([name@host](#)) is the one that allows the VO-DAI and VO-Dance to access the internal schema. If the SQL server runs on a different server machine than the other two tokens, it needs to be created accordingly. The name itself, *vodance*, can be customized, but take care of apply this change also inside the VO-DAI (see later) because this is also the user used by the two service samples.

VO-DIDB contains also credentials for the first access to VO-DAI (see later on), with an administration user (*admin*) and its password (*changeit*).

3.2. VO-Dance Administration Interface Setup

On a working Python/Django installation setting up the VO-DAI requires a few configuration steps. Unzip the tar.gz admin package (e.g. VODance_admin.tar.gz) in a folder of your choice, e.g. in the home directory of the user you're using for the full process. You'll end up with a folder, with name *VODance* that contains the admin site Django module, plus a couple of files: *django.conf* and *django.wsgi*. These two are needed by Apache (and its wsgi module) to correctly serve VO-DAI.

You can probably put these files wherever you like, but considering they are related to the HTTP serving you'll probably prefer having these two in a different place than the one that hosts the Django module. On the opposite, even if in principle the VO-DAI python django module may work from every location on your disk, it is suggested, for simplicity in deploying it, to put it under the *site-packages* python folder (the same where django gets installed).

Anyway, the only things to check and configure on these are:

- for the *django.wsgi* make sure that the *path* parameter points to the real full path of the *VODance* Django module (e.g. some /usr/lib/python2.7/site-packages/VODance);
- for the *django.conf* some more steps are needed:
 - make sure to import it using the *httpd.conf* directives (or copy it there);
 - modify the *<Directory>* directives accordingly to the actual paths you used (i.e. the VO-DAI Django module location and the *django.wsgi* location) and the related *Alias*;
 - be sure you create a symbolic link for the *static* folder, that is used for VO-DAI styling according to Django package, use (once you are in the same folder of the *django.wsgi* file) something like:

	VO-Dance Installation Guide	Document No.	VO-Dance-001
Issue/Rev. No.		0.3-1	
Date		December 2013	
Page		7 of 9	

```
$ ln -s /usr/lib/python2.7/site-packages/django/contrib/admin/static/admin/
```

After these steps you can point to your localhost in your browser, probably at `http://localhost/vodance/admin`, and check the results. Since you're missing some settings for the VO-DAI DB connection, you'll probably get an error in response. If it's a DB connection error it's highly probable that everything else is working properly and you can go on. If some error due to missing permissions arises, be aware that it could depend on Selinux configuration for httpd; this command (requires *root* permissions) may help:

```
$ chcon -v -R --type=httpd_sys_content_t {wsgi script folder} {VO-DAI folder}
```

DB configuration is found in the `settings.py` file in the VO-DAI module folder. Edit it and configure the `DATABASES` section. It could be that only the `PASSWORD` has to be changed to the one you decided when configuring the MySQL user. Change also the e-mail address for the admin user, save the file and exit.

Now, if everything is correct, you can point the browser to the localhost VO-Dance admin location and you'll get a login form. User is *admin* and the password is *changeit* (as it claims, it's better if you change it at first login). Once in the admin interface you will find the two sample services details and other things (that will be described in the VO-Dance User's Manual, not yet available). For the two services you need to edit them (just click on the name or id number to access the edit page), insert the MySQL password and save them. Otherwise VO-Dance web application will be unable to connect to the samples schema.

(One last thing is the change the temporary label on top of VO-DAI. To do so go to the VO-DAI module folder, inside the `templates/admin` subfolder, and edit the `block-branding` text, the one between `<center>` HTML tags)

3.3. Glassfish Configuration and VO-Dance Deployment

Now that you have the VO-Dance internal DB in place and the VO-DAI running and able to manage it, the last step is to deploy the actual web application that manages the services. To do so you only need to set up one pair of JDBC connection pool and resource in your Glassfish web container and deploy the `.war` VO-Dance package. Both these steps require connecting to the Glassfish Administration Console (probably on `localhost:4848`).

Firstly, add a new JDBC Connection Pool (the name can be anyone of your choice) with the following properties:

- *Resource Type: javax.sql.ConnectionPoolDataService*
- *Database Vendor: MySQL*

Clicking *next* enter then the following properties (after deleting all the automatically set up ones):

- `databaseName`
- `portNumber`
- `serverName`
- `password`
- `user`

	VO-Dance Installation Guide	Document No.	VO-Dance-001
		Issue/Rev. No.	0.3-1
		Date	December 2013
		Page	8 of 9

with values set accordingly to the MySQL configured user. Click *finish* and re-enter the connection pool you just created. On the main panel you'll find a *ping* button. If the pool is set up properly the result should be a *succeed*.

With the connection pool properly setup you can create the *JDBC Resource* itself. Add a new one: its name **MUST BE** *jdbc/VODanceData* (case sensitive) and will point to the newly created pool above. That's all, the pair pool-resource is ready.

Now you can go to the *Applications* Glassfish panel and click on *Deploy*, choose the *.war* file (e.g. *VODance.war*) follow instructions (mainly entering some description) and choose the *ContextRoot* for your application (it will be automatically set to the name of the *.war* package, but you'll probably prefer something like *ivoa* or other distinctive context name). Finalize the deployment.

You can now test and go productive with your VO-Dance instance.

4. Testing

There are mainly two ways to test your freshly installed instance of VO-Dance using the small set of data we provide with the installation package: browser test and VO client applications.

Here explain the basics to test with a browser.

Go to the *ContextRoot* of your VO-Dance new instance: should be something like *http://localhost:8080/your_context_choice*. HTTP port may be different if you have a non-default Glassfish setup. You should find an explanation to query the services.

The *vodance_samples* schema contains two services pre-defined as examples:

- a “cone” with id=2
- a “siap” with id=1

Try them. If you get *VOTable* in response (with *status OK* in the *<INFO>* element) everything is working fine and you can start working on your own services.

Appendix

A. Web Containers and Servers Port Numbers

Installing Apache and Glassfish on a server machine with no other web servers or web containers in place should be easy and straightforward, ending up with Apache httpd listening on port 80 and Glassfish web server listening on port 8080. If, however, other services are already using those ports or you want a different configuration, please take care of propagating the changes.

B. Apache HTTPD Configuration Notes

Apache configuration directives may change depending on the Apache version you are using. The provided *django.conf* files give you the syntax for versions 2.2 and 2.4. The web server version is usually reported on the welcome page.

Also consider that locations of configuration scripts are not fixed, so you'll probably need to investigate to properly finalize the configuration of your Apache-WSGI environment.

	VO-Dance Installation Guide	Document No.	VO-Dance-001
Issue/Rev. No.		0.3-1	
Date		December 2013	
Page		9 of 9	

C. Environment Versions

Java JDK versions 1.6 and 1.7 have been used throughout the VO-Dance development, maintenance and production phases. It has never been tested in earlier versions. Oracle (Sun) JDK have been used mainly because the Glassfish web container complains during installation if it finds another JDK, but there are no special issues that should prevent VO-Dance from working if you use openJDK (never properly tested, anyway).

VO-Dance Installation Package is built with Glassfish v.3 libraries. If different web container versions are needed you must contact IA2.

Apache 2, 2.2 and 2.4 have been used in various phases. Only Apache directive configuration changes are needed if using ≤ 2.2 or 2.4 versions. CentOS and Fedora provided Apache httpd packages as well as DjangoStack httpds have been successfully used. WSGI module for Apache is required by python-django.

MySQL 5.5+ and 5.6 have been used mainly, however also 5.1+ should work. Server, client, shared and devel MySQL packages are required. A JDBC connector is required for Glassfish Connection Pools and VO-Dance to work. MySQLdb python module is required to connect the VO-DAI django module to MySQL. This last MySQLdb requires python-devel.

Python 2.6.6 and 2.7 are known to work, never tested with python 3.

Django versions 1.5.4 and 1.5.5 work, also previous versions may work. Django 1.6 is not supported due to the usage of a deprecated feature in the VO-DAI module.