

# WADE INSTALLATION GUIDE

USAGE RESTRICTED ACCORDING TO LICENSE AGREEMENT.

Last update: 09-November-2013 - WADE 3.3

Authors: Giovanni Caire (Telecom Italia S.p.A)

Copyright (C) 2013 Telecom Italia

WADE (Workflows and Agents Development framework) is a software platform that facilitates the development of distributed multi agent applications where agent tasks can be defined according to the workflow metaphor. From a different point of view WADE can be seen as a Workflow Engine suitable to be used for business process orchestration.

This document gives a practical guide to install and configure the environment required to develop and run WADE based applications. It does not go into the details of WADE administration and fine-tuning and it does not provide indications about how to develop and run workflows. Refer to the Wade User Guide and Wade Development Tutorial for details on these topics.

The WADE library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, version 2.1 of the License.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Java and HotJava are trademarks of Sun Microsystems, Inc., and refer to Sun's Java programming language and HotJava browser technologies. The Workflows and Agents DEvelopment Framework WADE is not sponsored by or affiliated with SUN Microsystems, Inc."

## TABLE OF CONTENTS

1	DEVELOPMENT AND RUNTIME ENVIRONMENT .....	3
2	SETTING UP THE DEVELOPMENT ENVIRONMENT .....	3
3	SETTING UP THE RUNTIME ENVIRONMENT .....	6
4	ACTIVATING THE SYSTEM .....	14

# 1 Development and Runtime Environment

When creating a WADE based system two types of environment can be distinguished:

- **Development Environment.** This is the environment where developers work when creating the workflows and the other custom components (if any) that make up the system. Typically each developer in the development team has his own “private” environment running in his machine. Within this environment he is able to execute an internal instance of the system to try what he is doing in practice and debug it .
- **Runtime Environment.** This is the environment where the system actually runs. In many cases there are different instances of the runtime environment, for instance one for the functional tests another for the performances/stress tests and a third one for the actual production.

This guide describes how to setup both a development environment and a runtime environment and how to deploy artifacts created in the development environment on a runtime environment instance as outlined in Figure 1.

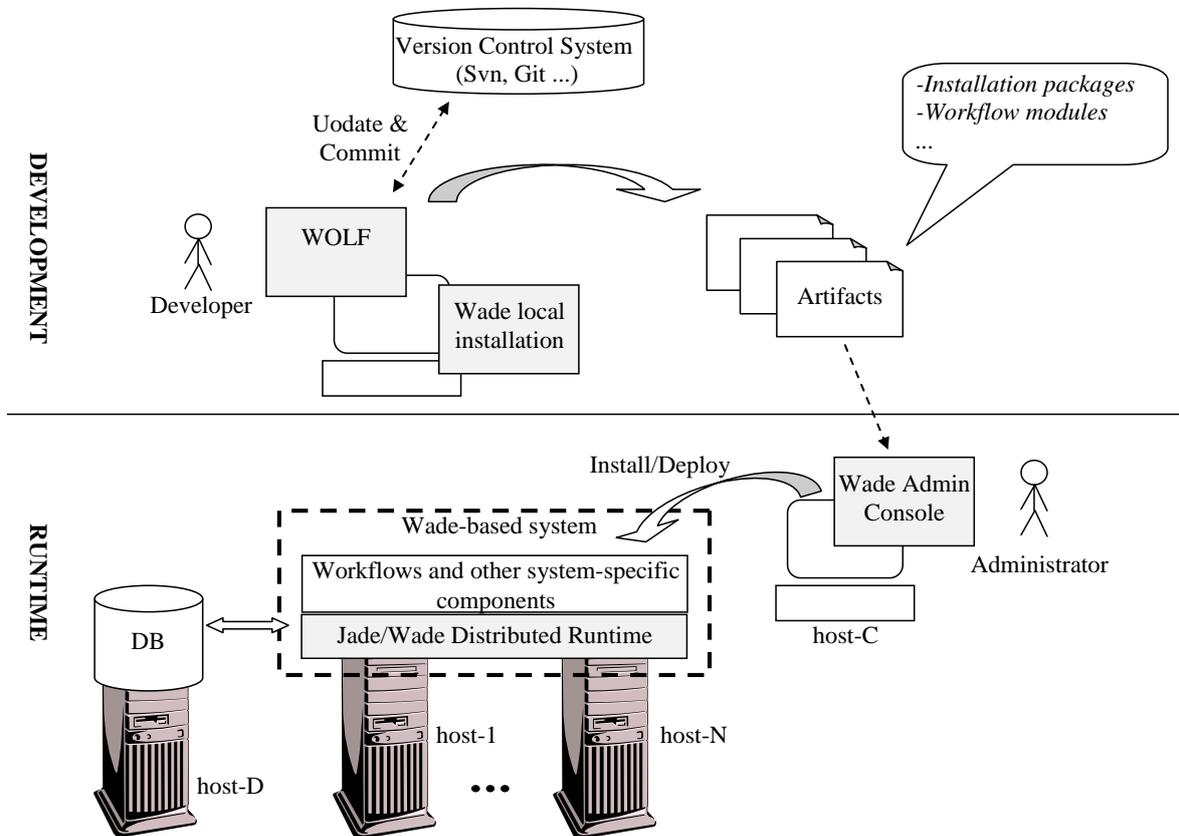


Figure 1. Development and Runtime environments

# 2 Setting up the development environment

The development environment for WADE based systems consists of the Eclipse IDE ([www.eclipse.org](http://www.eclipse.org)) plus WOLF, an Eclipse plug-in that provides support for all relevant WADE related development features. More in details WOLF allows

1. setting up an Eclipse project targeting the development of a WADE based system,
2. graphically creating and modifying workflows,
3. trying and debugging workflows and other components that make up the system and

4. producing the packages that can be installed/deployed in the runtime environment. Furthermore we do suggest keeping a local WADE installation. This is not strictly necessary for the development environment since WOLF embeds an internal WADE installation. However, especially if you plan to use some of the WADE Add-Ons (for instance the Persistence Add-On that is necessary to execute long-running workflows and to browse workflow executions by means of the Wade Administration Console) a stand-alone WADE installation is useful/required to try developed artifacts in practice.

Setting up the development environment is described in details in the Wade Development Tutorial. In this section we just summarize the main steps.

1) Unless already there install **Eclipse** (Helios **or later** versions). Refer to the Eclipse documentation for details about how to do this.

**NOTE:** WOLF uses two Eclipse plugins, GEF and DRAWD. Depending on the Eclipse distribution you are using such plugins may already be present in your Eclipse installation or not. In particular

- [Eclipse IDE for Java EE Developers](#) Both plugins included
- [Eclipse IDE for Java Developers](#) Both plugins included
- [Eclipse for RCP and RAP Developers](#) Both plugins included
- [Eclipse Standard](#) Plugins **NOT** included

If you are using an Eclipse distribution that does not include such plugins yet, you have to install them (Refer to the Eclipse documentation for details about how to do that).

2) Install the **WOLF** plug-in. WOLF can be downloaded from the download area of the WADE web site (<http://jade.tilab.com/wade>). Once downloaded, installing WOLF is as simple as unzipping the downloaded zip package inside the Eclipse root directory.

3) Install **WADE**. WADE can be downloaded from the download area of the WADE web site (<http://jade.tilab.com/wade>). Once downloaded installing WADE is as simple as unzipping the downloaded zip package somewhere on your disk thus obtaining a directory structure similar to that below.

```
.../
|--wadeSuite/
    |--wade/ ← We refer to this as the WADE home directory
        |--add-ons/
        |--bin/
        |--cfg/
        |--lib/
        |--projects/
        ...
```

4) If you need them, install the required **WADE Add-Ons** by downloading them from the download area of the WADE web site and unzipping the distribution zip packages in the same directory where the WADE distribution zip package was unzipped.

5) Create the **Eclipse project** where the sources of the system under development are going to be created. Within Eclipse, from the main menu select File → New → Other. In the resulting dialog box, select Wade Project as shown in Figure 2 and then select Next.

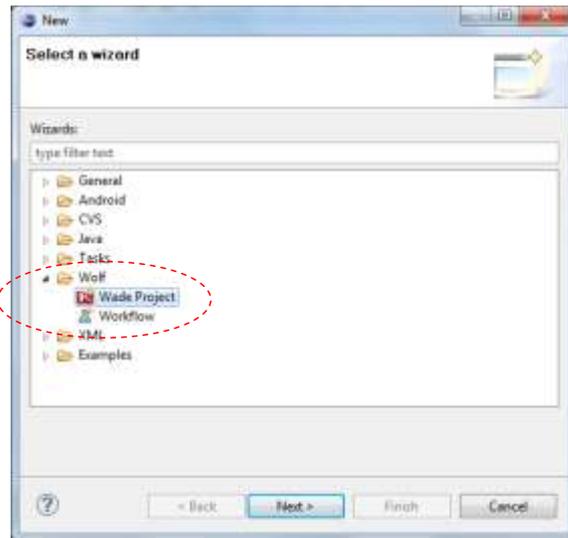


Figure 2. Creating a Wade Project

The New Wade Project dialog box appears. Type in the project name and click Finish.

6) **Configure the newly created Wade project.** Right-click on the project root and select Properties. Then select Wade Configurations as shown in Figure 3 [1]. In the General panel select Use an external WADE installation (Figure 3 [2]) and insert the home directory of your local WADE installation (see step 3).

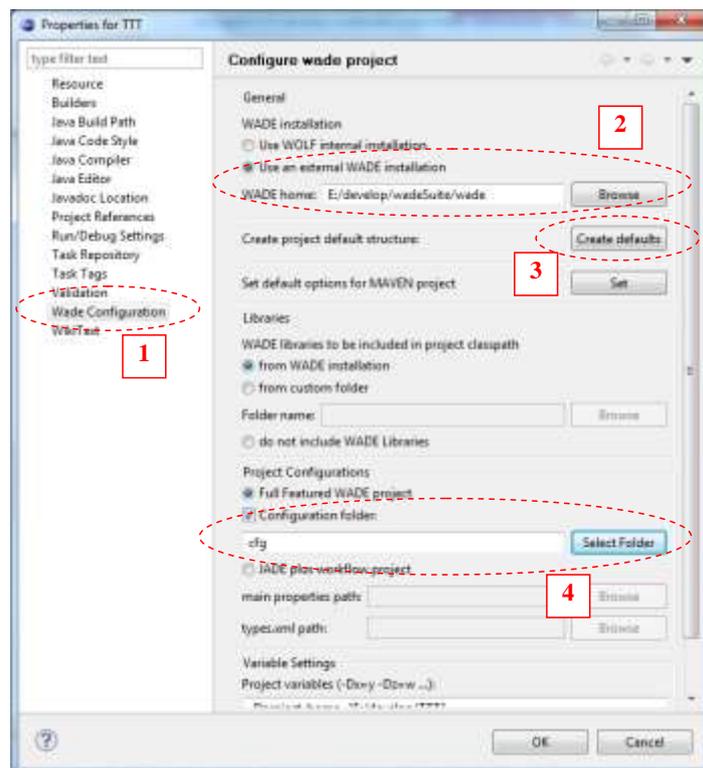


Figure 3. Configuring the project

Then, if you are starting a new project from scratch, let WOLF create the default project structure by clicking on the Create defaults button (Figure 3 [3]).

Finally (Figure 3 [4]) check the Configuration folder check box and select the configuration folder (cfg in the default project structure) i.e. the folder where all WADE-related configurations are located.

At this point you can start creating workflows and other components (if any) that make up the system. See the Wade Development Tutorial for more information about that.

If more than one WADE-based systems must be developed, repeat steps 5 and 6 for each system under development.

7) **Try the system locally.** As long as you develop the code of your WADE-based system, Wolf allows you to start it inside Eclipse (refer to the Wade Development Tutorial for details about that).

In certain cases however you may also want to start it outside Eclipse. To do that

- make a copy of the default `.properties` file inside the `projects` directory of your local WADE installation. Call it `<system-name>.properties`. Then open the `<system-name>.properties` file and edit the `project-home` and `project-classes` properties to reflect where your system is (full path) and where its classes can be found (relative path to the `project-home`).

- After that open a shell, go to the `bin` directory inside your local WADE installation and start the WADE BootDaemon

```
startBootDaemon
```

and the system Main Container

```
startMain <system-name>
```

- Having started the Main Container, the Management GUI should appear as shown in Figure 4.

This is a minimal GUI intended for easily trying the system inside the development environment only.

In the runtime environment we do suggest using the Wade Administration Console as will be described in next section.

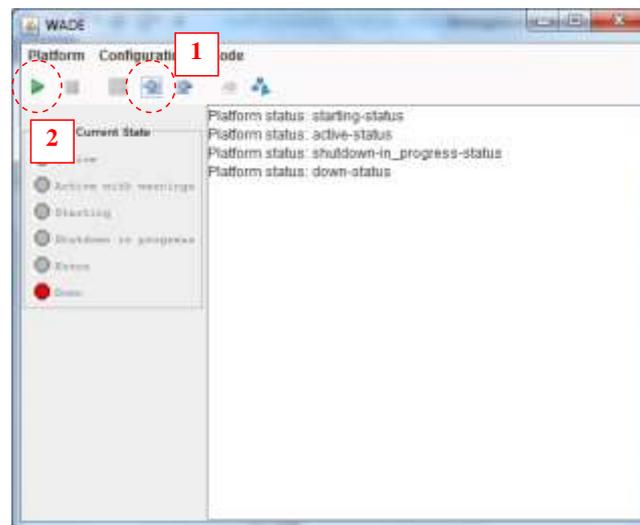


Figure 4. The Wade Management GUI

By means of the Management GUI, import the configuration defining which agents have to be started in your system (1) and finally activate the system (2).

### 3 Setting up the runtime environment

This section describes how to set up the runtime environment for a WADE-based system.

As depicted in Figure 1, a typical installation comprises one or more hosts (`host1` to `hostN`) where the system logics actually runs plus 1 host (`hostC`) for the Administration Console and 1 host (`hostD`) for the DB where WADE keeps all persistent information (e.g. those related to workflow executions). Of course **having dedicated hosts for the Administration Console and for the DB is NOT mandatory: when the computational load the system has to deal with is limited, all components can be installed in a single host.**

The Runtime Environment setup phase goes through the steps described hereafter.

**NOTE:** How to setup the DB is out of the scope of this document. Refer to the documentation of the DBMS of your choice for details about that. WADE has been successfully tested with Oracle, MySQL and H2, but, since it uses Hibernate and standard SQL, we expect it to work with other DBMS too.

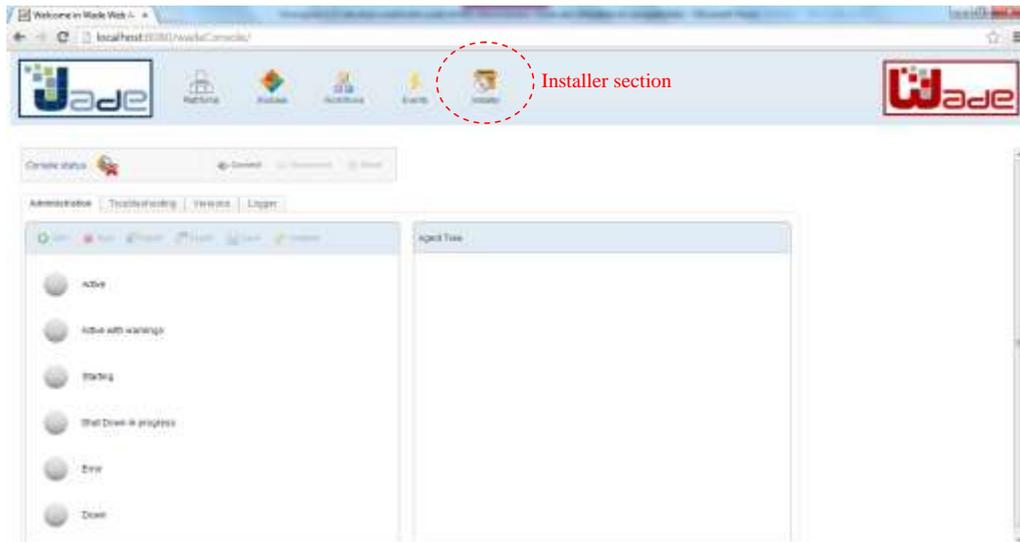
### 1) Administration Console set up (hostC)

The Wade Administration Console is a Web Application (.war) that can run in whatever Web Container such as Apache Tomcat. Therefore

- Install a Web Container. Refer to the documentation of the selected Web Container for details about how to do that.
- Unzip the content of the WADE Web Console (just the wadeConsole.war file) in the webapps directory of the Web Container installation.
- Start the Web Container (again refer to the selected Web Container documentation for details about how to do that). As a result, the wadeConsole.war file is expanded in the directory structure below.

```
<Web Container home>/
|--...
|--webapps/
    |--...
    |--wadeConsole/
        |--...
        |--WEB-INF/
            |--classes/
            |--lib/
            |--platforms/
            |--hostsConfigurations/
            |--labels/
            |--console.properties
            |--...
```

- To check that everything is OK open a Web Browser and type the <http://hostC:8080/wadeConsole> url (this url assumes you are using Apache Tomcat as Web Container and you started it on the default port 8080). The Wade Administration Console should appear as depicted in Figure 5



**Figure 5. The WADE Administration Console**

- The `console.properties` file highlighted in bold includes the configurations required by the Administration Console to connect to the Main Container of the system that is going to be installed. We suggest opening it and setting the `multiplatform-mode` to `true`.

```
# Multiplatform mode
multiplatform-mode=true
```

Depending on the version of the Wade Administration Console you are using the `multiplatform-mode` parameters may already be set to `true`. To detect that, it is sufficient to see if a combo box labelled “Select a platform” is displayed just above the Console Status panel.

In general this configuration allows supporting more than one WADE based system by means of a single Administration Console. Furthermore it hides the configuration steps needed to connect the Administration Console to the successive installations of the system that will be done in steps 6 and 7.

NOTE: The Web Container must be restarted for this configuration to take effect.

## 2) Installer Daemon setup (host1 – hostN)

Considering that one of the main characteristics of WADE is distribution, often a WADE-based system runs on several hosts (typically for scalability reasons). In order to facilitate the installation of both WADE and project specific components on all hosts, a small background process called Installer Daemon must be installed and activated on each host.

- The Installer Daemon can be downloaded from the download area of the WADE web site (<http://jade.tilab.com/wade>). Once downloaded, installing it is as simple as unzipping the downloaded zip package somewhere thus obtaining the following directory structure

```
...
|--wadeInstaller/
    |--bin/
    |--cfg/
    |--...
```

- To activate the Installer Daemon go to the `wadeInstaller/bin/` directory and type

```
nohup ./startInstallerDaemon.sh &
```

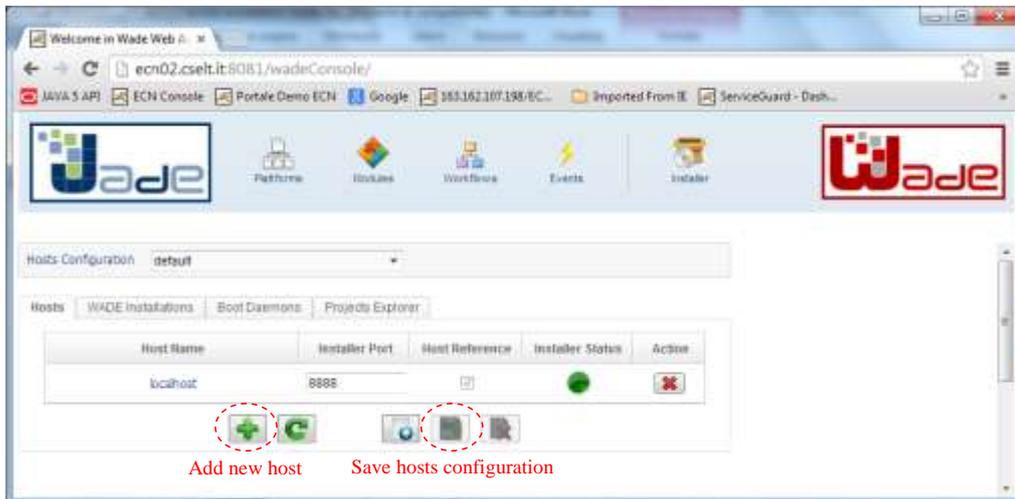
The above command assumes you are working in a Linux environment. Similar `.bat` scripts are provided for a Windows environment.

This step must be repeated for each host where the WADE based system you have to install/execute must run.

From this point on all successive steps to install and successively update both WADE and system specific components can be performed from the `Installer` section (highlighted in Figure 5) of the Wade Administration Console.

### 3) Hosts declaration

Before the actual installation operations can take place it is still necessary to declare all hosts (`host1 – hostN`) where such operations will be performed in the Administration Console. To do that click on the `Installer` section (see Figure 5). The Hosts Configuration tab should appear as depicted in Figure 6.



**Figure 6. The Hosts Configuration tab**

By clicking on the `Add new host` button you can populate the hosts table with the relevant hosts. Each row refers to a host and more in details

- The `Installer Status` column shows whether or not the `Installer Daemon` is active on that host. If you declare all and only hosts where the `Installer Daemon` was installed and activated in step 2, you should see all green icons.
- The `Host reference` column allows declaring the host that will be considered as reference in all successive installation and maintenance operations to detect which versions of WADE are actually installed.

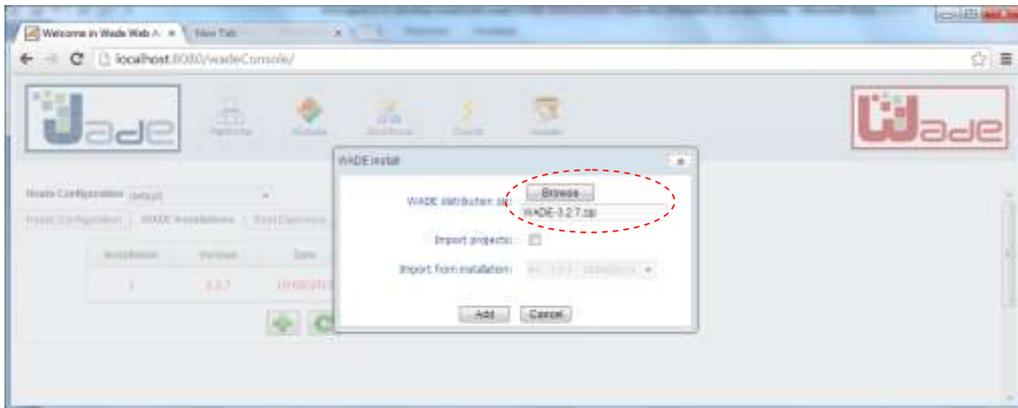
Once done, the hosts configuration must be saved by means of the `Save hosts configuration` button.

### 4) WADE Installation

At this point we are finally ready to actually install WADE. To do that

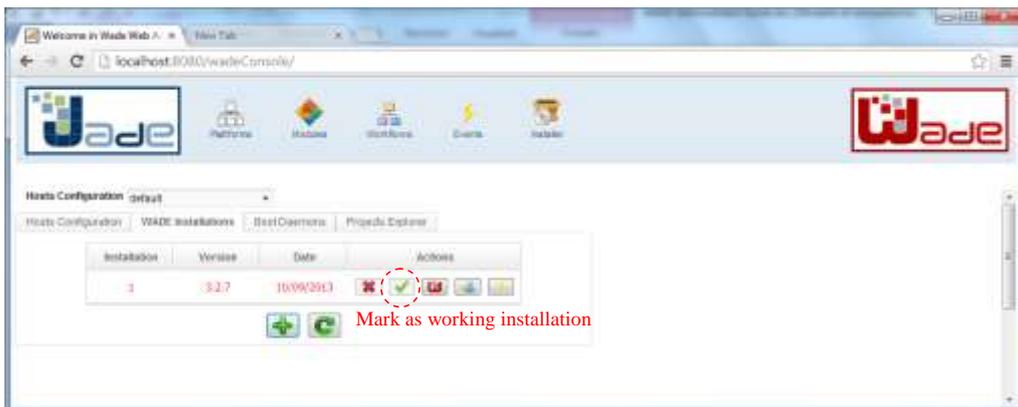
- Download the WADE installation package from the download area of the WADE web site (<http://jade.tilab.com/wade>) and store it somewhere on your local disk.

- In the **Installer** section of the **WADE Administration Console** select the **WADE Installations** tab and click on the **Add new WADE Installation** button. In the dialog box represented in **Figure 7** use the **Browse** button to select the **WADE installation package** you just downloaded. Finally click **Add**.



**Figure 7. WADE Installation package selection**

A dialog box reporting the result of the installation in all hosts declared in step 3 is shown.  
 - A new row appears in the **WADE Installations** tab representing the new **WADE installation**. Mark it as the **“working installation”** by means of the button highlighted in **Figure 8**.



**Figure 8. The WADE working installation**

Since we’ve just set up the environment, at this point we have a single **WADE installation**. However in general there may be more than one (e.g. following the installation of a successive **WADE release**). The **working installation** identifies the **WADE installation** to which all successive administration operations refer to.

Just for information purposes, let’s take a look now at what the **WADE installation process** produced. Open a shell in the host marked as *reference* in step 3 (the situation should be exactly the same on all other hosts) and go to the directory where the **Installer Daemon package** was unzipped in step 2: this is how the situation should look like now.

```

...
|--wadeInstaller/ ← This was produced in step 2
    |--bin/
    |--cfg/
    |--...
|--wadeRepository/ ← This was produced in step 4
    
```

```

|--working.properties  ← Keeps info about working installation
|--wade_1_yyyymmdd/    ← The home directory of installation 1
    |--bin/
    |--cfg/
    |--lib/
    |...

```

## 5) WADE Add-Ons installation

In case you need one or more WADE Add-Ons such as the Persistence Add-On, install them as described below.

- Download the Add-On installation package from the download area of the WADE web site (<http://jade.tilab.com/wade>) and store it somewhere on your local disk.
- In the Installer section of the WADE Administration Console select the WADE Installations tab and click on the Install WADE Add-On button. As in previous step use the Browse button to select the Add-On installation package you just downloaded.
- Repeat the above operations for all required Add-Ons.

## 6) Project (System) Registration

In general more than one WADE-based system can run on the same WADE platform at the same time. Each system is seen by the platform as a “*Project*“. As a consequence the first phase to install the system specific software is to register a project. To do that select the Projects Explorer tab in the Installer section of the WADE Administration Console and click on the Add new project button as shown in Figure 9.

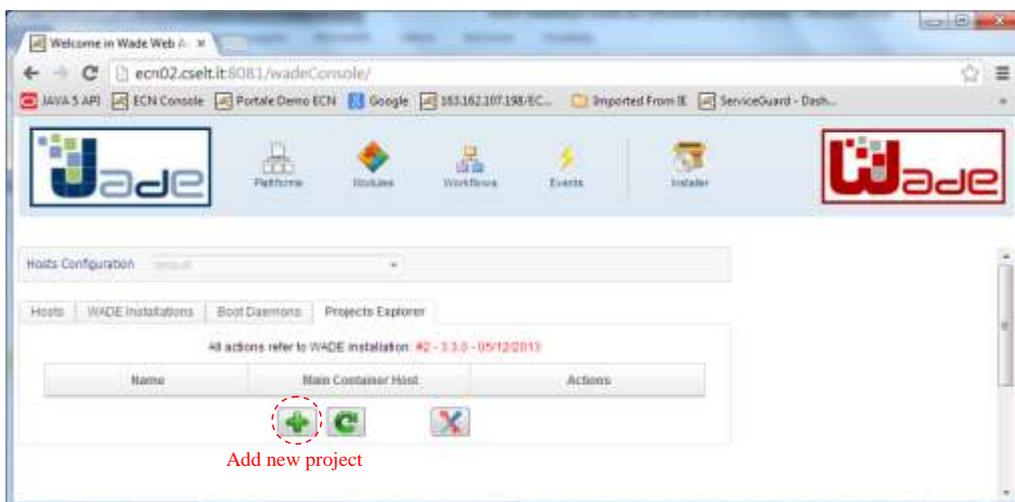


Figure 9. The Projects Explorer tab

In the Project setup dialog box that appears type the **project name** and the directory where all releases of the system corresponding to this project will be installed. This directory can be expressed as a relative path (*in this case it is relative to the Installer Daemon running directory* `../wadeInstaller/bin`) or absolute path. A good suggestion is to use `../../<system-name>`.

In case different installation directories must be used in one or more of the hosts declared in step 3, click on the Configure Project hosts button. Finally click on the Add button.

**NOTE:** If the indicated installation directory does not exist, the Installer Daemon automatically creates it. If it already exists, the Installer Daemon checks that it is empty. If this is not the case an error message is shown. This is because, if at a later step the project will be removed, the

installation directory will be fully deleted and this would delete any pre-existing stuff included in that directory too.

A new row appears in the `Projects Explorer` tab representing the newly registered project. To complete the registration phase, in the `Main Container Host` column indicate the host where the Main Container of the system will run as shown in Figure 10.

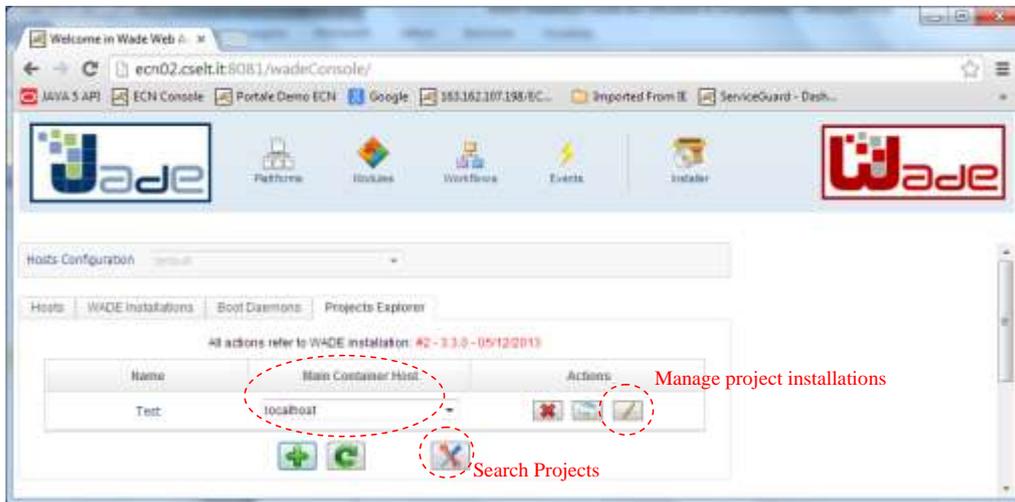


Figure 10. A new project in the Projects Explorer tab

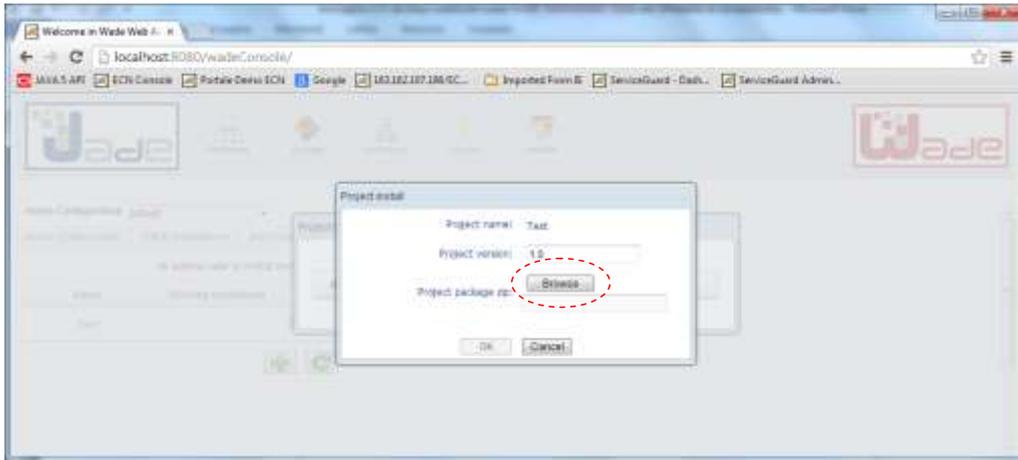
**NOTE: If you did not install everything from scratch as presented in this document, but you installed the WADE Console 3.1 or later in an environment where WADE and one or more WADE-based systems where already installed, the Project Explorer tab will likely be empty the first time you connect. In order to “recover” your projects click on the Search Projects button.**

### 7) Project (System) installation

The final step of the Runtime Environment setup procedure is the actual installation of the system specific software. Before doing that it is necessary to produce the system installation package. This is typically done by the development team by means of Wolf (refer to the Wolf documentation for details).

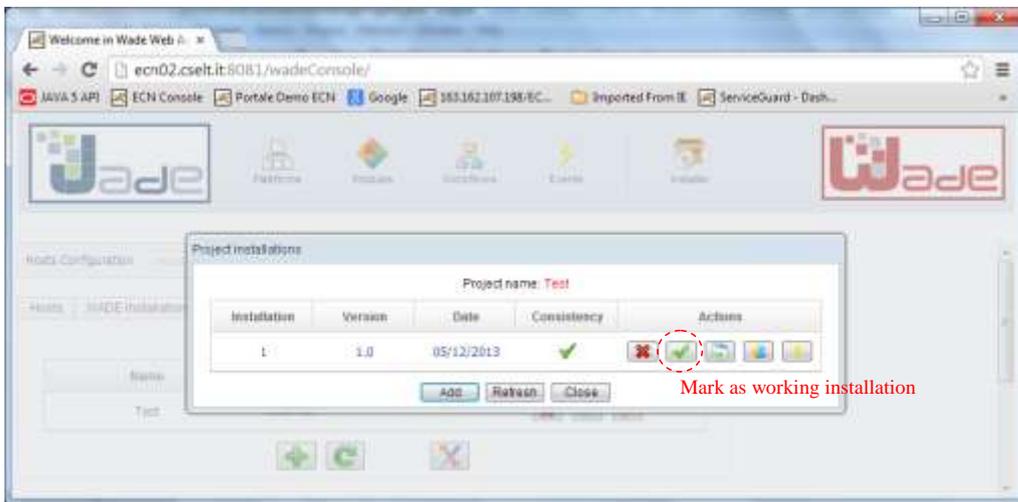
Once the system installation package has been produced and stored in your local machine, click on the `Manage Project Installations` button highlighted in Figure 10. In the `Project Installations` dialog box that appears click on the `Add` button.

In the `Project Install` dialog box shown in Figure 11 insert the project version and then use the `Browse` button to load the installation package. Finally click `OK`.



**Figure 11. The Project Install dialog box**

A new row appears in the Project Installations dialog box representing the new installation. Mark it as the “*working installation*” as highlighted in Figure 12.



**Figure 12. A new installation in the Project Installations dialog box**

Just for information purposes, let’s take a look now at what the Project Installation process produced. Open a shell in the host marked as *reference* in step 3 (the situation should be exactly the same on all other hosts). Assuming you used `../../<system-name>` as project installation directory, go to the directory where the Installer Daemon package was unzipped in step 2: this is how the situation should look like now.

```

...
|--wadeInstaller/ ← This was produced in step 2
  |--bin/
  |--cfg/
  |--...
|--wadeRepository/ ← This was produced in step 4
  |--working.properties
  |--wade_1_yyyymmdd/ ← The home directory of installation 1
    |--bin/
    |--cfg/
    |--lib/
    |--...

```

```

|--<system-name>/ ← This was produced in step 7
  |--<system-name>_1_yyyddd/ ← Installation 1 project home
    |--cfg/
    |--classes/
    |--lib/
    |--log/
    |--deploy/
    |--...

```

NOTE: The content of the directory of installation 1 of your system may slightly vary depending on what you put in the system installation package.

## 4 Activating the system

Having followed the installation procedure described in Chapter 3, the system can now be activated as described below.

### 1) Activate the Boot Daemons.

The Boot Daemon is a small process that must run in all hosts where the system logics runs (host1 to hostN) and is responsible to activate the actual system components (wade containers with system specific agents on top) at startup. Refer to the WADE User Guide (<http://jade.tilab.com/wade/doc/WADE-User-Guide.pdf>) for more details about the Boot Daemon and in general for a detailed description of the WADE architecture.

To activate the BootDaemon in all hosts select the `Boot Daemons` tab in the `Installer` section of the WADE Administration Console and click on the `Start all boot daemons` button.

### 2) Activate non-WADE components

Activate non-WADE component (if any) needed by the system to run (e.g. a DBMS). How to do that is out of the scope of this document.

### 3) Activate the system Main Container

Open a shell in the host indicated as *Main Container Host* in step 6 of the installation phase. Go to the home directory of the WADE working installation and then move to the `bin` subdirectory

```

...
|--wadeInstaller/
  |--.../
|--wadeRepository/
  |--working.properties
  |--wade_1_yyyymmdd/
    |--bin/
      |startMainContainer.bat
      |startMainContainer.sh
      |--...
    |--.../

```

Launch the Main Container by typing (the command line below refers to a Linux environment)

```
nohup ./startMainContainer.sh <project-name> &
```

where <project-name> is the name given to the project in step 6 of the installation procedure.

#### 4) Activate the whole system

In the WADE Administration Console move to the Platforms section and then select the your system in the Select a platform combo box. Then click on the Connect button as highlighted in Figure 13. Once connected, the Status panel (as well as the Agent Tree panel) should become enabled and should show that the system is Down. Click on the Start button and wait for the system to become Active.

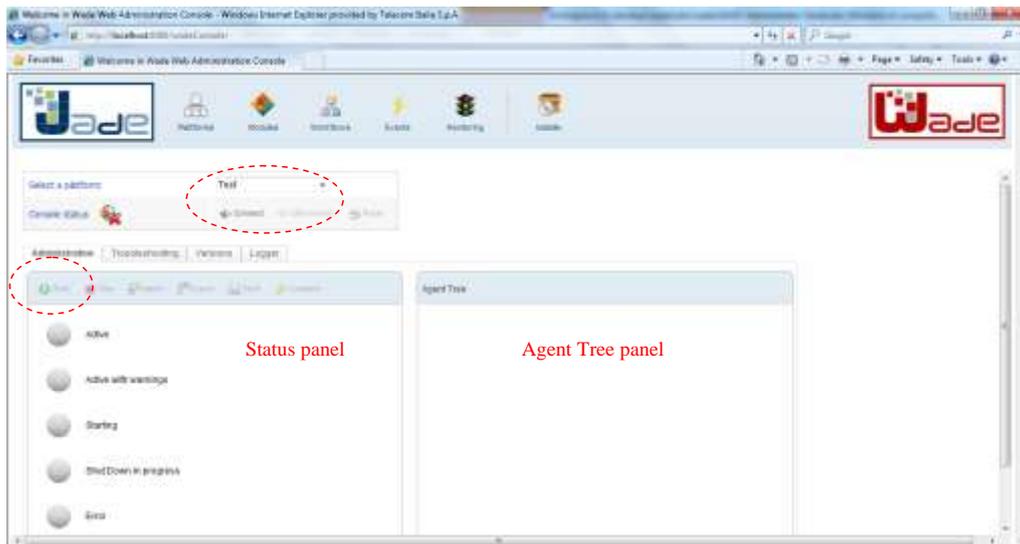


Figure 13. The Platforms section

**NOTE:** If the Select a platform combo box does not appear, it means that you did not set the multiplatform-mode parameter in the Administration Console configuration properties as explained in step 1 of chapter 3.