

# NSRA instalation

This guide assumes that OpenEdge Architect will be used to work with NSRA, instalation steps not needed for other environments are specifically noted.

The first step is to install OpenEdge10.1C onward, to be able to run the NSRA server, including the client preprocessing module ('*cliente*' module - '*client*').

## Server code instalation

Unzip **nsra\_server-X.zip**, where X can be '*devel*', '*test*', '*prod*', '*time*' for development, test, production and response time measurement environments, respectively. This file must be unzipped on the OE Architect workdir.

*NOTE: if you haven't run OE Architect yet, run it for it to create the working dirctory in which to unzip the files.*

*NOTE: technically it can be unzipped anywhere, as long everything is "pointed" to the right directory..*

Once unzipped the **nsra\_server** folder, with all the server structure inside, will be created. This structure includes the '*bases*' folder that contains the structure and data for the '*nsradb*' and '*cliente*' databases, the first required by the server to work, and the second required by the '*cliente*' module.

## Databases creation

Start OpenEdge '*DataAdministration*' by using menu option '*OpenEdge → Admin → Data Administration*' (inside OE Architect).

*NOTE: if the option is grayed, is due to the licenses installe, run it independtrly.*

*NOTE: this application can be run in other environments too, including character Linux environments.*

Inside database admininstrator, create an empty database and load the structure (**nsra.df**) and content (**nsradb** directory, inside **bases** directory) for nsradb database.

Repeat the process for the '*cliente*' database.

Once databases are created, start '*Progress Explorer Tool*' (or '*OpenEdge Explorer*') for the creation of the database servers to serve these databases on some computer port..

*NOTE: databases can be served on any computer, not neccesarilly on the development workstation, they can even be shared among all the workstations..*

As the first step (if OpenEdge instalation is recent) you must add an OpenEdge server, to doit use the context menu that is shown by right-clicking over the '*Progress Explorer*' *node*, on the left tree. On the showed window specify the data for '*Server*' and '*User*' (usually '*localhost*' for server and anything for user, without a password). Leave the rest of the data with offered defaults and complete the creation, you can also check the checkboxes for conecting at startup and user information storage.

Once an OpenEdge server is available, connect to it (if shown with a red dot). By connecting a list of child nodes will be shown, one of which is '*Databases*', right-clic on it to show its context menu where you should choose '*New..*'.

Use the name you wish for the database server (same name as served database is recommended), enter the database full path and check (optionally) the checkbox for the database server to automatically start. Press 'Ok' and a new node will get shown inside 'Databases' folder, expand it until 'defaultServerGroup' node is shown and select the 'properties' context menu option for it, set a value for the 'Service name or port number' field (the first field of the properties).

Repeat the previous steps for each database to be served. If you want to use the Sports2000 demo you have to create a sports2000 database.

Read OpenEdge documentation for further details on database servers configuration.

*NOTE: once created, database servers, remain stopped, you have to start them manually.*

## **Project creation and configuration**

Inside OEA (OpenEdge Architect) create a new OpenEdge project ('New → OpenEdge Project') and set its name to **nsra\_server** (or whatever name you give to the NSRA server folder), make sure that the 'Location' field shows the full path to the folder containing the NSRA framework.

Press 'Next' button twice, until you get to the database connections configuration screen. Use the upper 'link' to create two database connections, one for **nsradb** and one for **cliente**.

Inside database connections screen use 'New' to create a new connection, set the connection name (**nsradb** is recommended) and set as physical name the real name of the database you have previously created. On the 'Logical name' set **nsradb** and fill 'Host' and 'Port' with the desired values for the connection.

Press 'Next' and uncheck the creation of an SQL link for the connection (this is not necessary but, in general, the SQL connection is not used and consumes resources). Press 'Next' again and then 'Finish'.

*NOTE: if you didn't create a database server you can check the 'Auto-start database server' checkbox, setting the port to use. DON'T check this checkbox if you already created a database server.*

Repeat the process for the **cliente** database (and sports2000 if used, with **sports** logical name).

Once connections are created, press 'OK' button, now you will see all the connections. Check the checkboxes and press 'Finish'.

Now you should see, on the left, all the folders that make up the NSRA server framework.

## **Module to develop creation**

Now you can start application development, by creating a new folder, inside the **nsra\_server** project, for the new module.

Inside the module directory you should create the same folder structure the other modules have:

**dao**  
**fabricas**

**dso**  
**informes**  
**plantillas**  
**servicios**  
**tablas**  
**utiles**

*NOTE: remember that if you are using derived exception classes, this classes should be created in an **excepciones** folder, inside the directory containing the class that uses this exception class.*

*NOTE: folder names can be changed to suit your language, except **dso**, **informes** and **plantillas**.*

## Version control systems

A CVS project (or a project of any other version control system) is normally used for version control of each project module.

If the module you are working on is already created, use the menu option '*New → Project*' to add it to the **nsra\_server** project, inside the first shown window choose '*Project from CVS*'. Create the repository connection and choose '*Use an existing module*', on the next window, to see all the CVS project found on the repository.

Choose the right project and press '*Next*', choose the creation of the sandbox inside an existing project ('*Check out into an existing project*') and press '*Next*'. Choose **nsra\_server** project and press '*Finish*'.

Now you should see the new folder for the module, already linked to the CVS repository.

If you don't have a CVS project for the new development, you will have to create one at some point, once you have, at least, the right directory structure for the module, right-click the module root folder and choose '*Team*' option and '*Add to version control*' option inside of it. Read Eclipse and the version control system help to finish the process.

## WebSpeed configuration (for web execution)

You need to configure one WebSpeed broker for services and, optionally, one for the preprocessing of client application (**cliente** module). This is only necessary to be able to run services, and get the client files, as it will happen in production environments. The server can be tested “manually” without needing WebSpeed.

To configure WebSpeed the '*Progress Explorer Tool*' program is used, inside '*WebSpeed*' folder you can change **wsbroker1** configuration (or you can create a new one), choose '*properties*' inside the broker context menu (by right-clicking with the mouse).

*NOTE: all this configuration changes can be made by manually modifying *ubroker.properties* file.*

First field to change is '*Working Directory*', that has to be filled with the full path to the NSRA server root folder (**nsra\_server** folder). In the '*Agent*' node, change the '*Agent startup parameters*' field, by putting:

```
-pf "<full path to nsra_server>\nsra\servicios\agent.pf" -weblogerror
```

Finally modify 'Environment Variables' node, setting this values:

Name	Value
LOG_PATH	<full path to nsra_server>\nsra\servicios\log
LOG_TYPES	*
Protocol	NSXML
RootDir	<full path to nsra_server>
SUPER_PROC	nsra/servicios/nsiniagnt.p

When you configure the **cliente** module broker (name it **wscliente**), some additional environment variables must be setted, and some values changed (it is important to change the broker port, to an available port). The field '*Agent startup parameter*' must be:

**-pf "<full path to nsra\_server>\cliente\agent.pf" -weblogerror**

This are the environment values to set:

Name	Value
LOG_PATH	<full path to nsra_server>\nsra\servicios\log
LOG_TYPES	*
Protocol	CLIENTE
RootDir	<full path to nsra_server>
SUPER_PROC	nsra/servicios/nsiniagnt.p
ClientRootDir	<full path to nsra_server>\xul

*NOTE: WebSpeed brokers must be started manually after creation. If 'AutoStart' checkbox isn't checked you will have to manually start the broker every time you need it.*

*NOTE: if you use different names for the brokers, you have to change the file **index.html**, inside the **xul** directory. You may have to change some xul files (search and replace 'wsbroker1').*

Before starting the brokers, edit all the **agent.pf** files you specify on the brokers configuration. Those files need to reflect the right database connection strings, verify each connection string and set the **H** and **S** parameters in order to connect the right databases.

## Web server

You can test NSRA server “manually” by using '*servicios.p*' program and creating a procedure file to execute, or an xml file to process, but if you want to try the **cliente** modulo or the interaction between client and server, you need to install a web server to access WebSpeed brokers.

Intallation of Apache web server is recommended, it is free and simple to configure, the next instructions have been used to install Apache 2.2.

Once Apache is installes, copy the **cgiip.exe** program, located on the **bin** directory of the

OpenEdge installation (usually C:\Progress\OpenEdge\bin), to the **cgi-bin** directory on the Apache installation directory.

Edit **http.conf** file, located on the **conf** directory, inside the Apache install dir, add one line:

```
Include conf/sitios/*.conf
```

Create a **sitios** directory, inside **conf** directory (on the Apache install dir) y dentro de ésta copie el archivo **nsra.conf** que se encuentra dentro de la carpeta **apache**, en el directorio **nsra\_server**. Edite **nsra.conf** y modifique ambas rutas para que contengan la ruta completa a **nsra\_server**.

Reinicie Apache, ahora debería poder acceder al demo Sports2000 utilizando una URL como la siguiente:

<http://localhost/nsra/>

al escribir la URL anterior, se mostrará el listado del directorio **nsra\_server**, seleccione el enlace a **xul** y, si su navegador soporta la interfaz xul, accederá al cliente xul con una URL como la siguiente:

<http://localhost/cgi-bin/cgiip.exe/WService=wscliente/cliente/servicios/cliente.p?params=,nsra.es,xul>