

Installing Wide Studio under Ubuntu Linux (Intel x86 version)

Written by: Lian Wen Wei

The Wide Studio Software packages for Ubuntu can be downloaded from the following site:
<http://packages.ubuntu.com/warty/devel/widestudio>.

In this article, we present the steps that are needed to get the Wide Studio Software running under Ubuntu Linux.

The Procedure

First, we have to download the packages needed for the installation. These Packages are listed below.

- 1) libmysqlclient10_3.23.56-2_i386.deb
- 2) libpq3_7.4.5-3ubuntu0.5_i386
- 3) libstdc++5_3.3.4-9ubuntu5_i386.deb
- 4) libwidestudio-image.deb
- 5) libwidestudio_3.50-2-2_i386.deb
- 6) libwidestudio-mysql_3.50-2-2_i386.deb
- 7) odbcinst1_2.2.4-9_i386.deb
- 8) unixodbc_2.2.4-9_i386.deb
- 9) libwidestudio-odbc_3.50-2-2_i386.deb
- 10) libwidestudio-perl_3.50-2-2_i386.deb
- 11) libwidestudio-pgsql_3.50-2-2_i386.deb
- 12) libwidestudio-python_3.50-2-2_i386.deb
- 13) libwidestudio-ruby_3.50-2-2_i386.deb
- 14) libwidestudio-dev_3.50-2-2_i386.deb
- 15) widestudio_3.50-2-2_i386.deb

Download these packages to the /tmp directory.

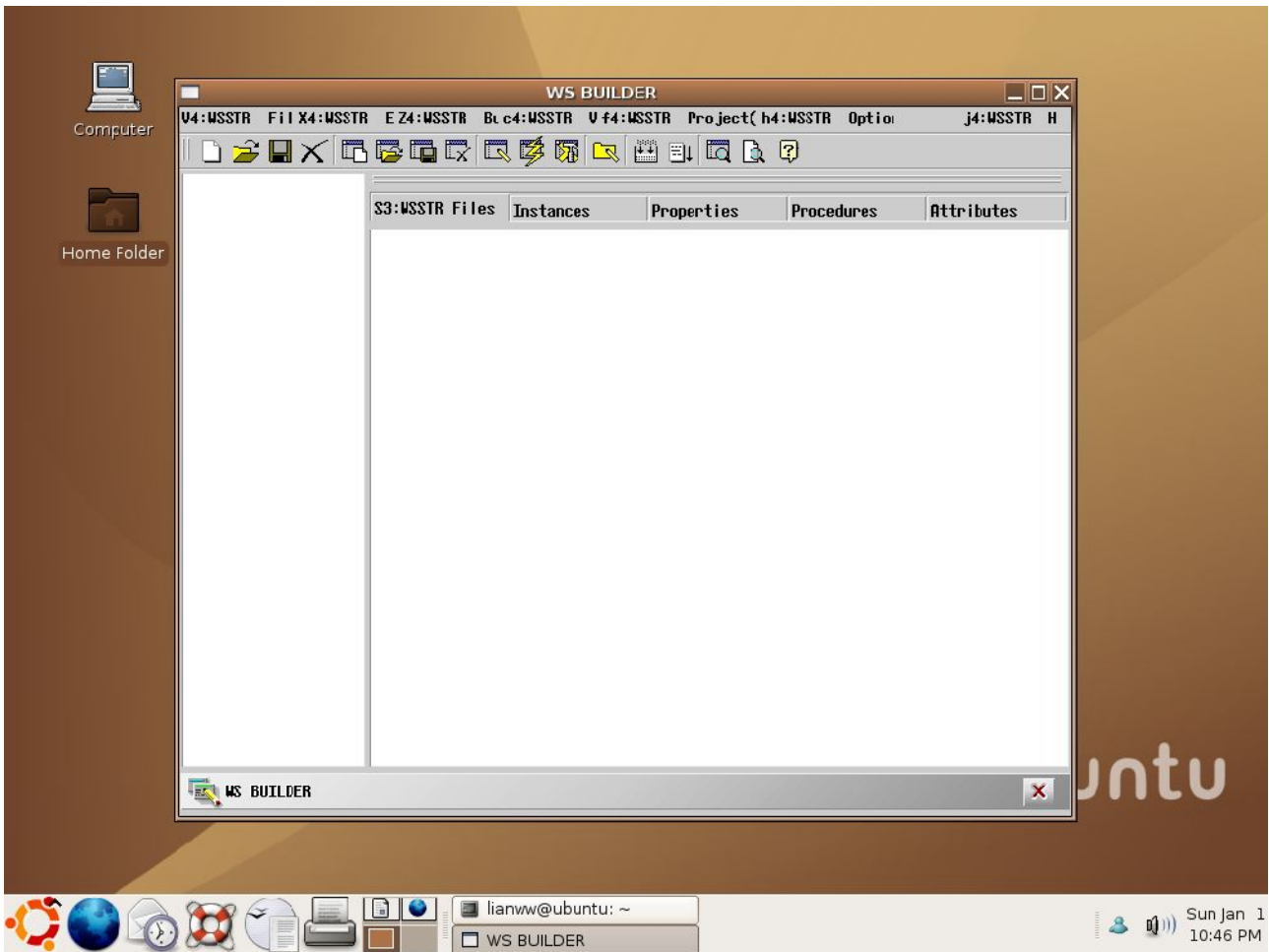
Change the directory to /tmp by typing “cd /tmp”

Go to the user root, by typing “sudo -s” follow by your password.

To unpack the packages, type “ sudo dpkg -i thepackagename.deb ”.

Note** you have to unpack the packages in the sequence listed above (from 1 to 15) to avoid broken package error.

Then, type “wsbuilder” to run Wide Studio. It will look some thing like this below.



Please note that this article is in progress and has not been fully verified by TE@Onghu. The completed article will be uploaded once it is ready. This version is only being provided for your reference. If you have any comments or notice any errors, please mail mohits@onghu.com so that your comments can be included in the final version.

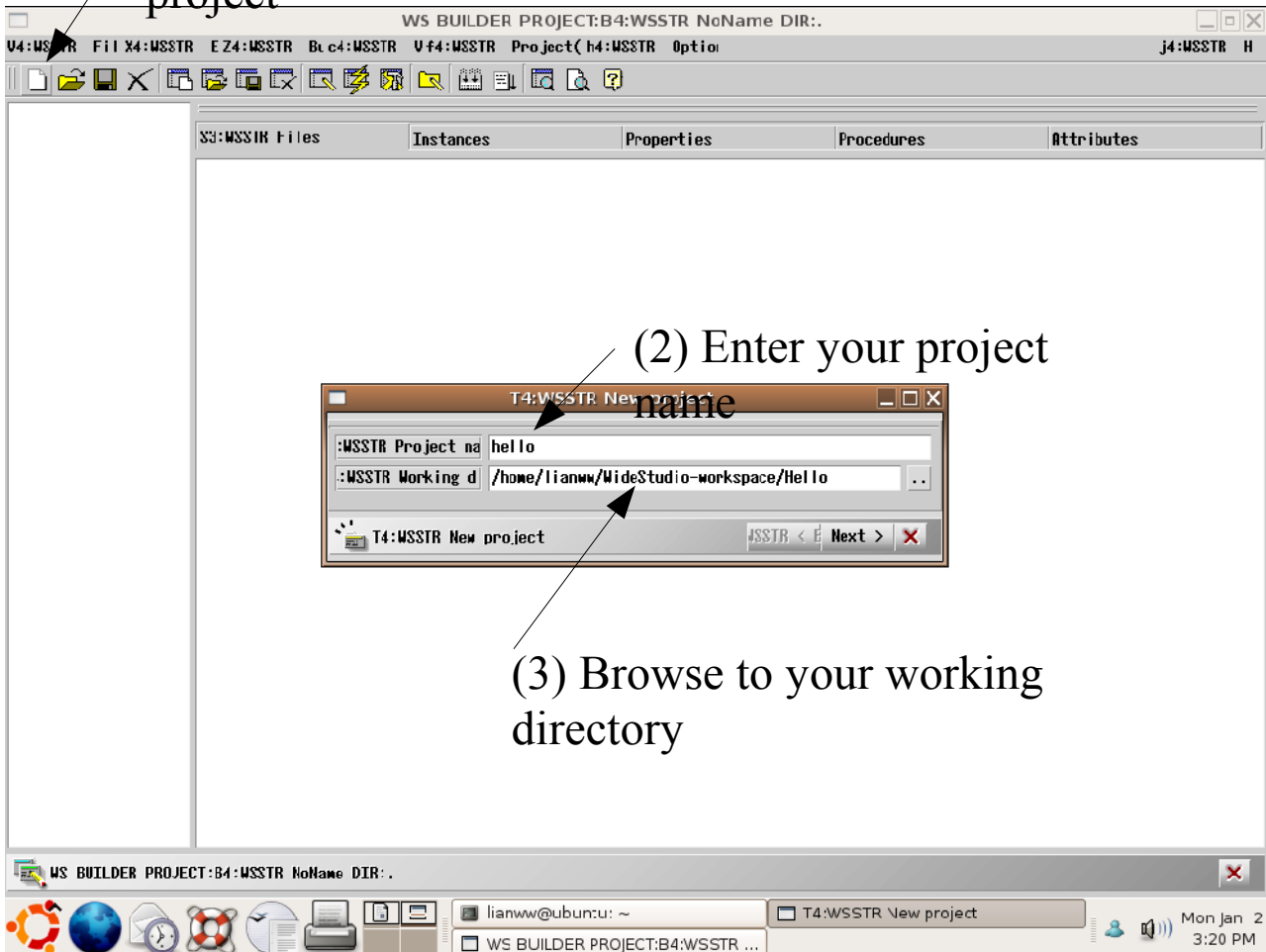
© Lian Wen Wei, 2006.

Create a hello project using Wide Studio running on Ubuntu Linux

To create a project “Hello”

To create a project for the "Hello" application. You just have to follow the steps below.

(1) Click on this icon to create new project



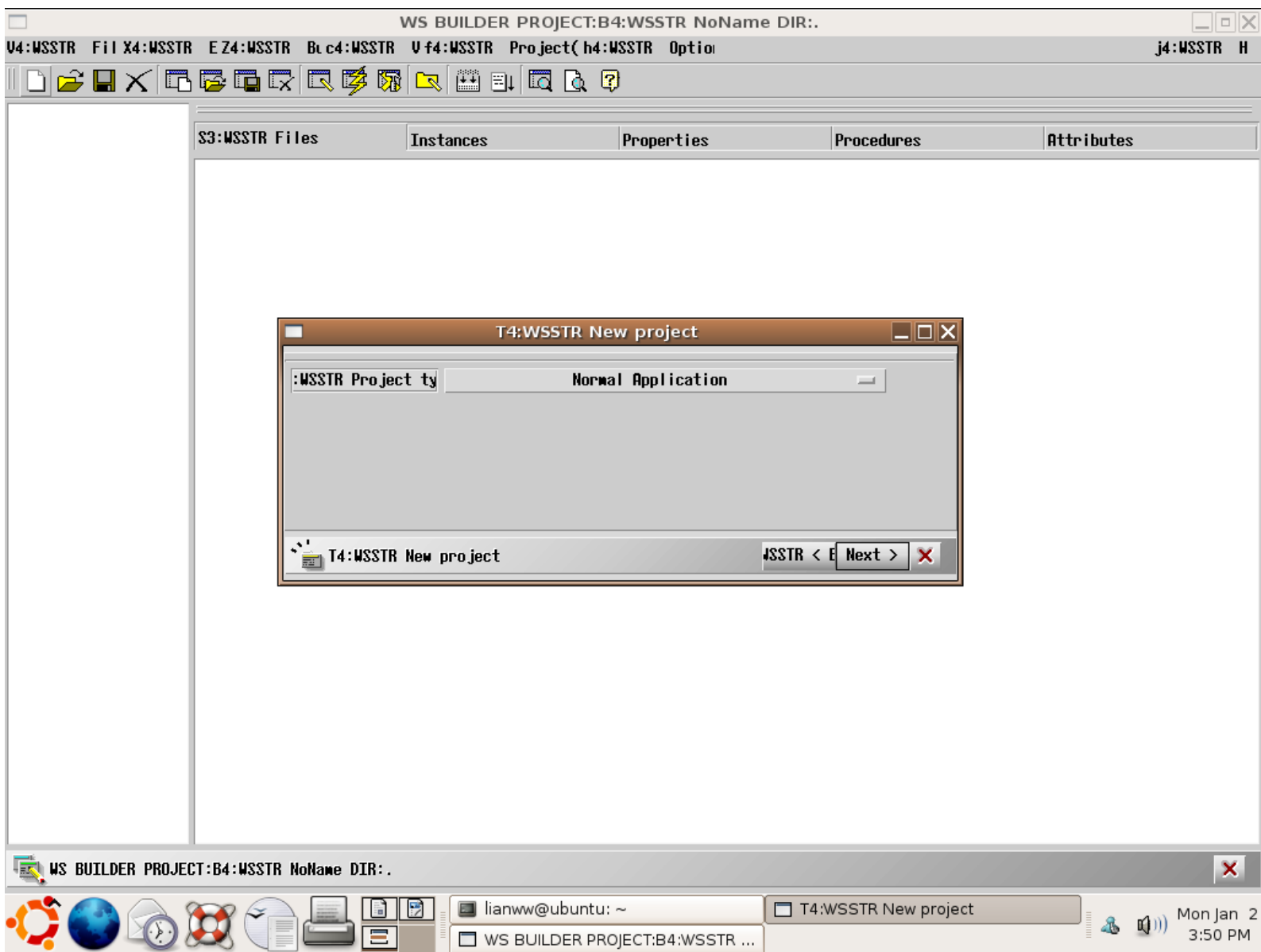
(2) Enter your project name

(3) Browse to your working directory

Please note that this article is in progress and has not been fully verified by TE@Onghu. The completed article will be uploaded once it is ready. This version is only being provided for your reference. If you have any comments or notice any errors, please mail mohits@onghu.com so that your comments can be included in the final version.

© Lian Wen Wei, 2006.

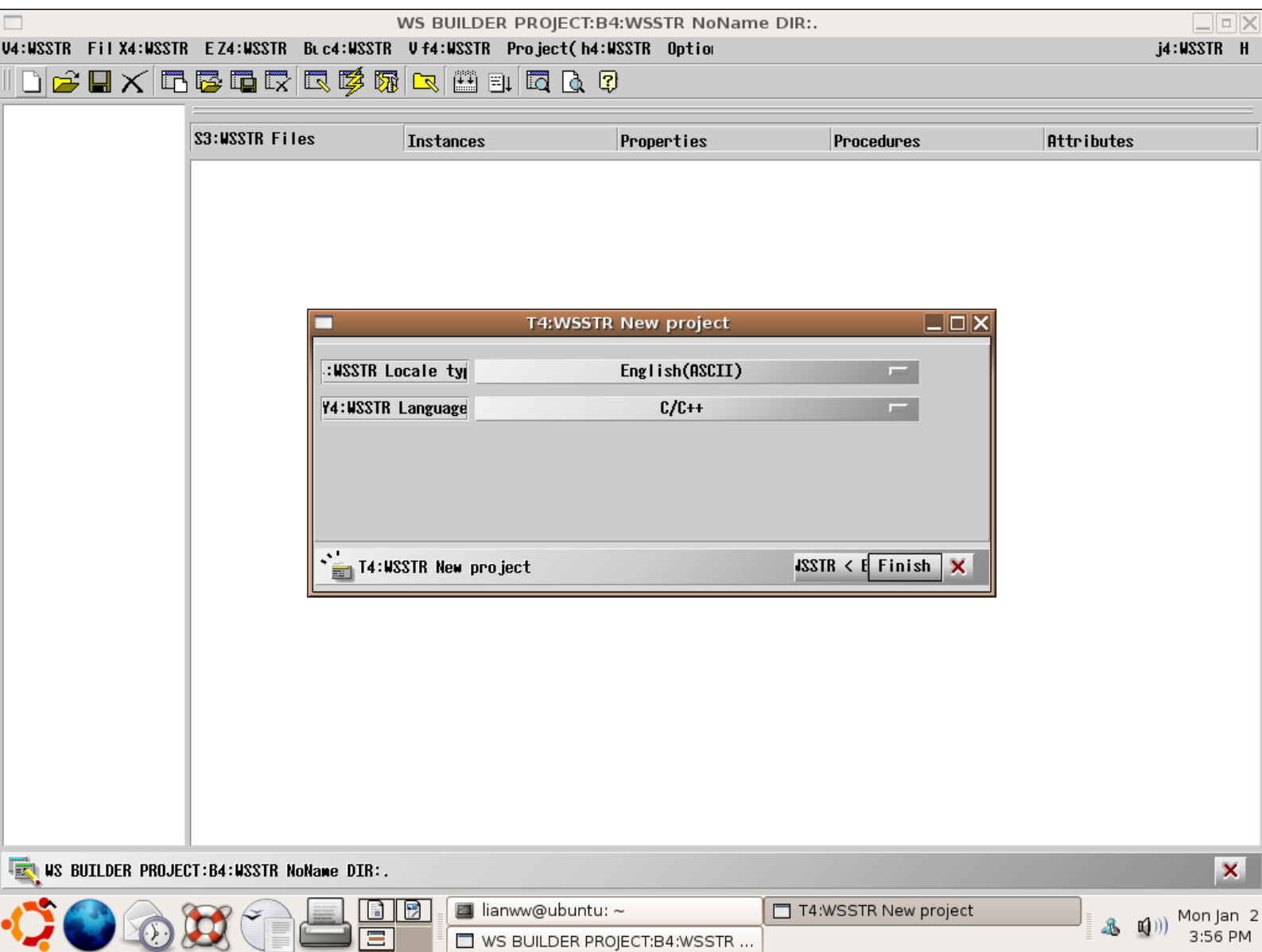
Click Next to proceed. You will come to a screen below.



There a number of project type to choose from they are the Normal Application, Class Library and Netscape Plugin.

Please note that this article is in progress and has not been fully verified by TE@Onghu. The completed article will be uploaded once it is ready. This version is only being provided for your reference. If you have any comments or notice any errors, please mail mohits@onghu.com so that your comments can be included in the final version.
© Lian Wen Wei, 2006.

In this case we are choosing a Normal Application.
Click Next to proceed.
You will come to a screen below.



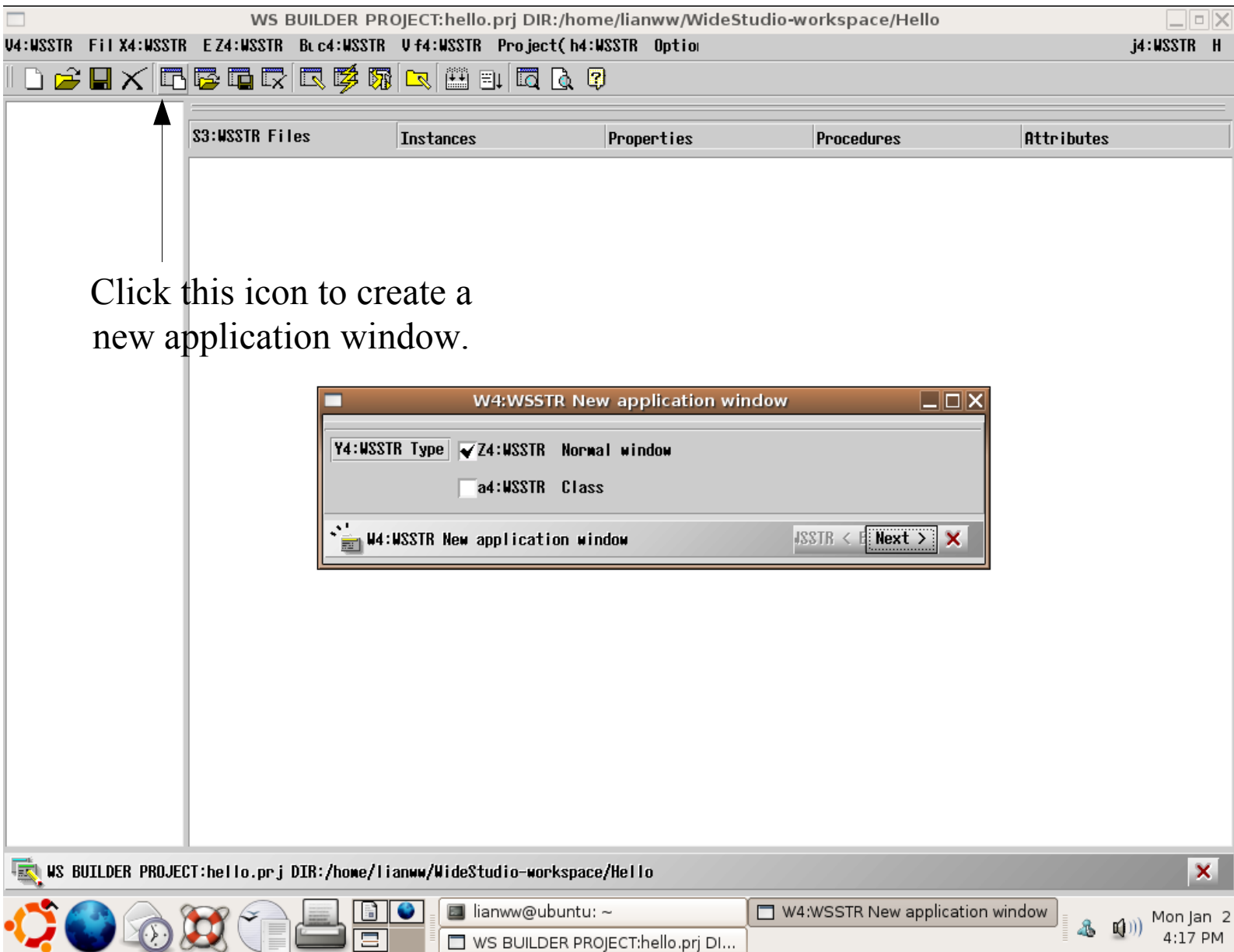
Please note that this article is in progress and has not been fully verified by TE@Onghu. The completed article will be uploaded once it is ready. This version is only being provided for your reference. If you have any comments or notice any errors, please mail mohits@onghu.com so that your comments can be included in the final version.
© Lian Wen Wei, 2006.

There are also a number of programming languages for the user to choose from. They are C / C++, Ruby, Python and Perl.

In our case, we choose English (ASCII) as our locale type and C / C++ as our programming language.

Click Finish to complete.

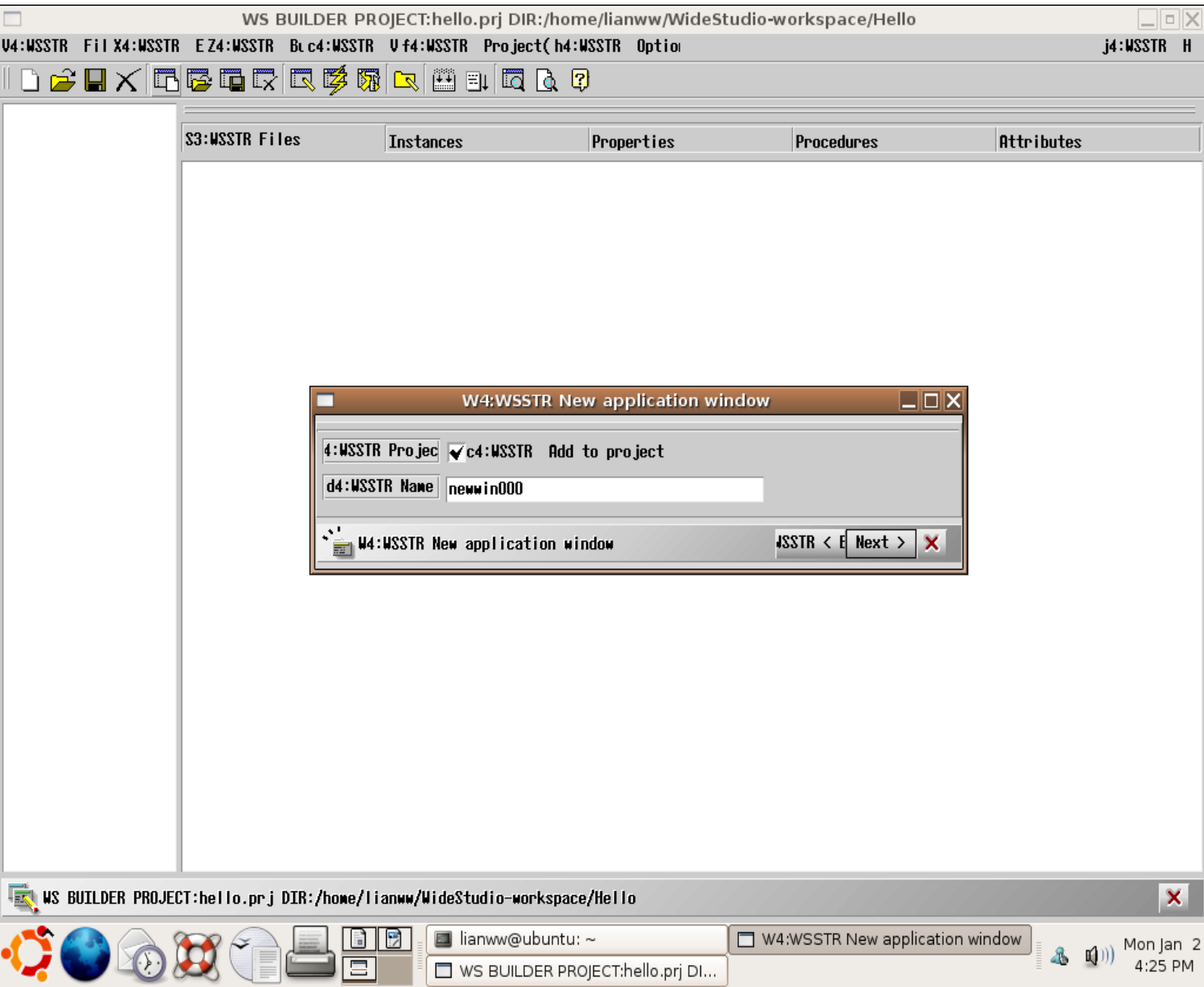
To create a new application window



In the new application window form, there are two options to choose from. They are the Normal Window and Class.

In our case, we choose the Normal Window option.

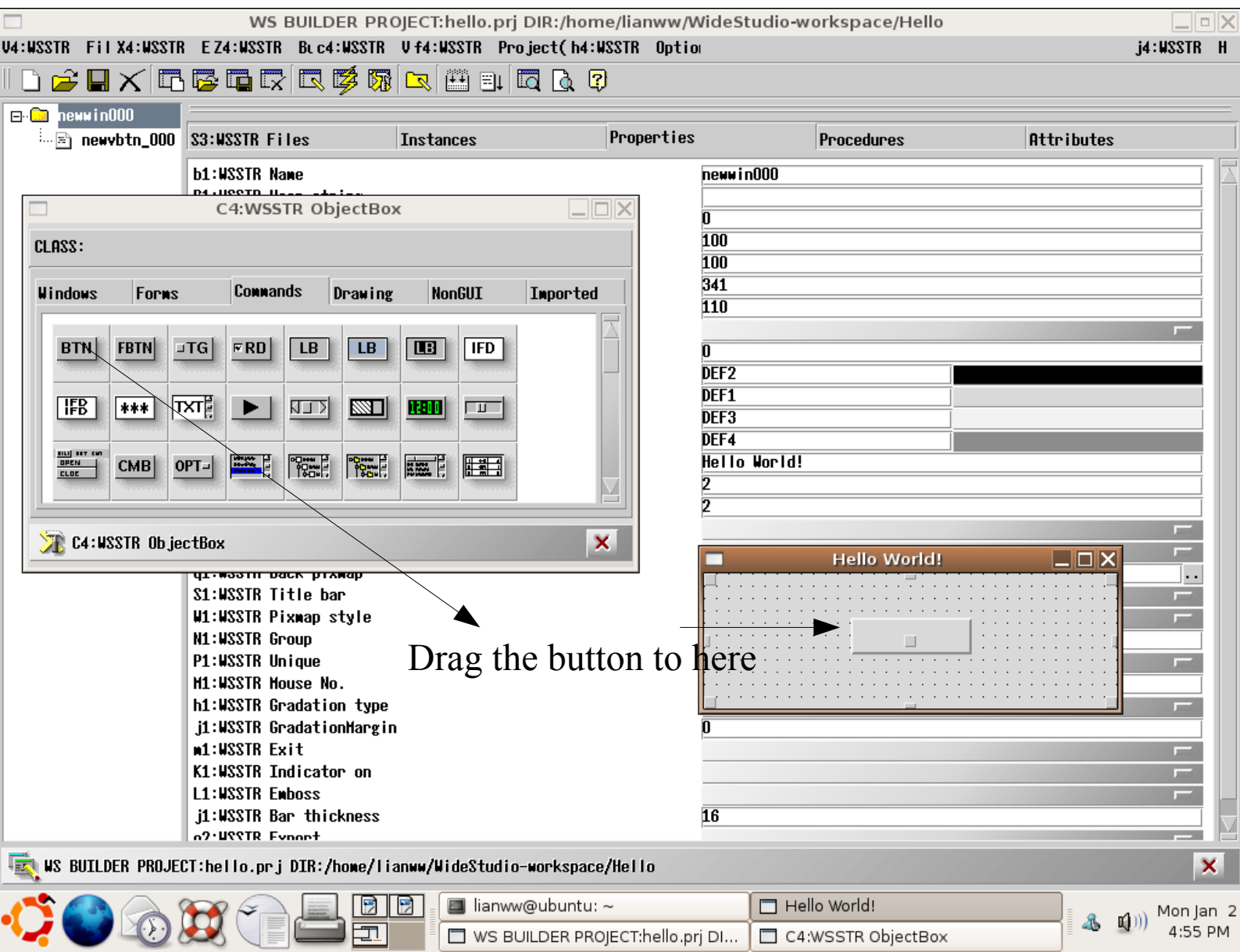
Click Next to proceed.

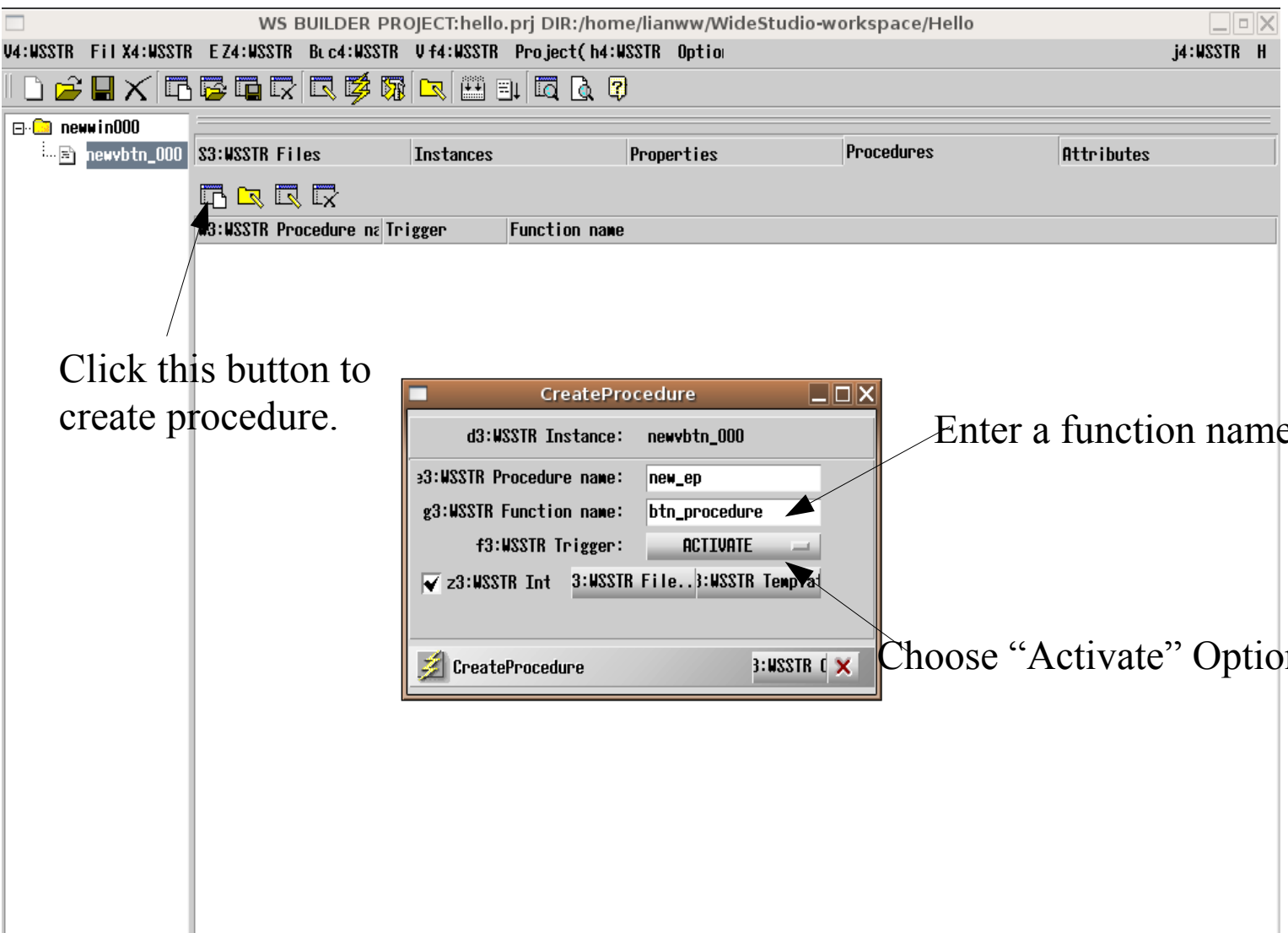


Click this icon to display the Object Box Form

The screenshot displays the WS BUILDER software interface. The main window title is "WS BUILDER PROJECT:hello.prj DIR:/home/lianww/WideStudio-workspace/Hello". The interface includes a menu bar, a toolbar, and a main workspace. A "C4:WSSTR ObjectBox" dialog box is open, showing a "CLASS:" section with tabs for "Windows", "Forms", "Commands", "Drawing", "NonGUI", and "Imported". The "Forms" tab is selected, displaying various form components like "MAIN", "OK", "Cancel", "Warning", "Error", "Info", "Help", and "Exit". Below the dialog box, a list of properties is visible: S1:WSSTR Title bar, W1:WSSTR Pixmap style, M1:WSSTR Group, P1:WSSTR Unique, M1:WSSTR Mouse No., h1:WSSTR Gradation type, j1:WSSTR GradationMargin, w1:WSSTR Exit, K1:WSSTR Indicator on, L1:WSSTR Emboss, j1:WSSTR Bar thickness, and a2:WSSTR Export. In the background, a "Hello World!" dialog box is also visible, showing a grid pattern. The system tray at the bottom shows the user "lianww@ubuntu: ~", the application "Hello World!", and the date "Mon Jan 2 4:44 PM".

To add a button to the application windows





Click this button to create procedure.

Enter a function name

Choose "Activate" Option

The screenshot shows the WS Builder IDE interface. At the top, the title bar reads "WS BUILDER PROJECT:hello.prj DIR:/home/lianww/WideStudio-workspace/Hello". Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Tools", "Documents", and "Help". A toolbar contains icons for New, Open, Save, Print, Undo, Redo, Cut, and Copy. The main workspace is divided into several panes. On the left, a tree view shows a folder "newwin000" containing a file "newbtn_000". Below this, a table lists procedures:

WS3:WSSTR Procedure name	Trigger	Function name
new_ep	ACTIVATE	btn_procedure

An arrow points from the text "Double click on new_ep" to the "new_ep" entry in the table. To the right, a Gedit window titled "btn_procedure.cpp (~/WideStudio-workspace/Hello) - gedit" is open, displaying the following code:

```
#include <WScom.h>
#include <WSCfunctionList.h>
#include <WSCbase.h>
//-----
//Function for the event procedure
//-----
void btn_procedure(WSCbase* object){
    //do something...
    object->setProperty(WSNlabelString, "World!");
}
static WSCfunctionRegister op("btn_procedure", (void*)
btn_procedure);
```

An arrow points from the text "G editor will appear" to the Gedit window. Another arrow points from the text "Add this statement to the code" to the line `static WSCfunctionRegister op("btn_procedure", (void*) btn_procedure);` in the Gedit window. The status bar at the bottom of the Gedit window shows "Ln 9, Col 45" and "INS".

Double click on new_ep

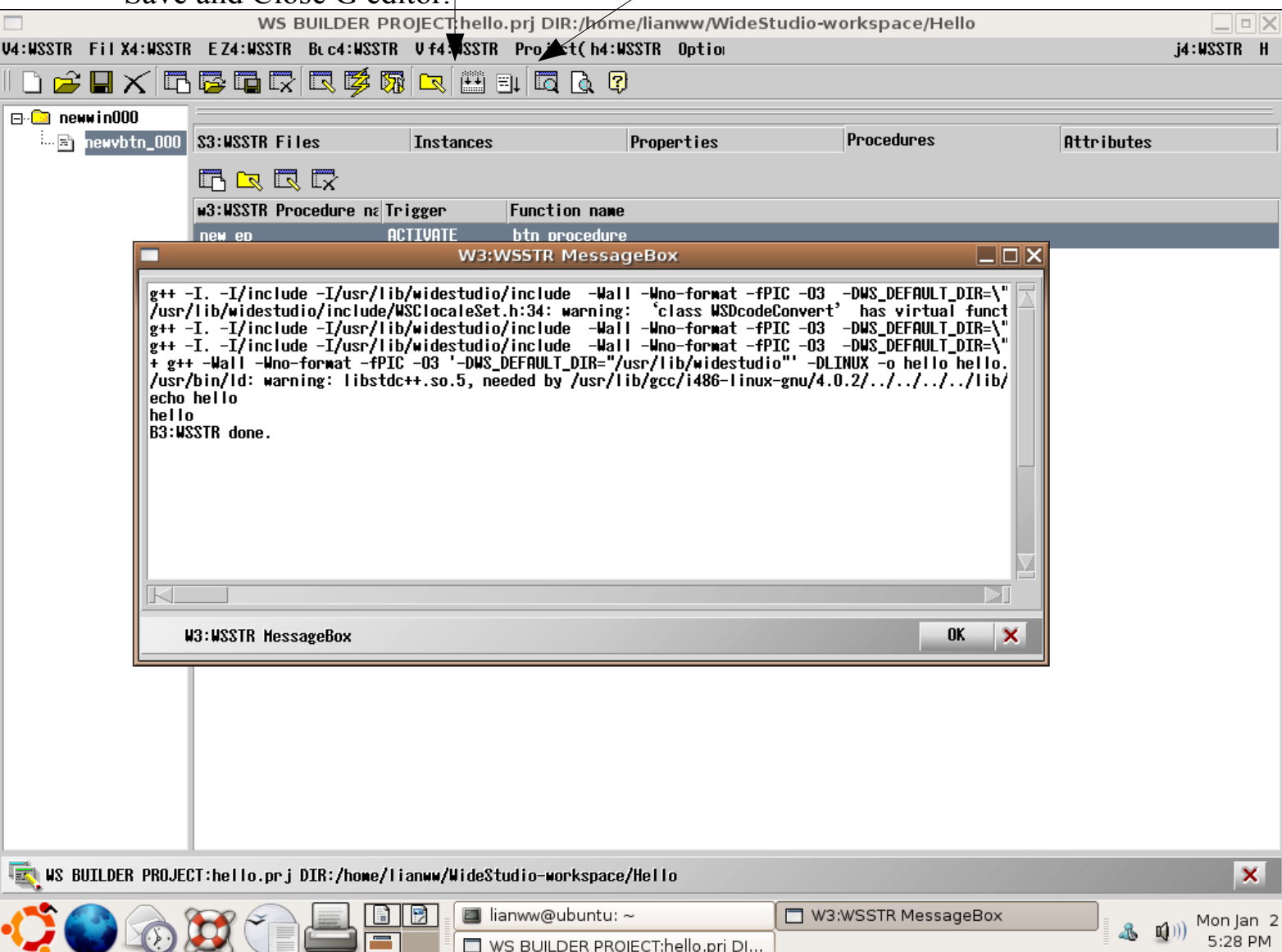
G editor will appear

Add this statement to the code

Click this icon to build the project

Click this icon to execute the project

Save and Close G editor.



The output of the Hello application

Please note that this article is in progress and has not been fully verified by TE@Onghu. The completed article will be uploaded once it is ready. This version is only being provided for your reference. If you have any comments or notice any errors, please mail mohits@onghu.com so that your comments can be included in the final version.

© Lian Wen Wei, 2006.



newwin000
newbtn_000

S3:WSSTR Files	Instances	Properties	Procedures	Attributes
----------------	-----------	------------	------------	------------

W3:WSSTR Procedure name	Trigger	Function name
new_ep	ACTIVATE	btn_procedure



newwin000
newbtn_000

S3:WSSTR Files	Instances	Properties	Procedures	Attributes
----------------	-----------	------------	------------	------------

W3:WSSTR Procedure name	Trigger	Function name
new_ep	ACTIVATE	btn_procedure



