



**Wide Studio Builder
User's Guide**

Contents

A Introduction	1
1 What is WideStudio?	1
2 What is C++ language?	2
3 Through WideStudio development	2
4 The multi-platform environment	3
5 The multi encoding	4
6 The open source license	6
7 Downloading of WideStudio	7
8 Building of WideStudio and installing it	8
8.1 Building and installing on the UNIX system	8
8.2 Installing with the binary rpm package on Linux operating system	9
8.3 installing on Windows operating system	9
8.4 Setting of the environment variables on Windows	11
8.5 Setting of the environment variables on the UNIX system	11
8.6 The other files on the UNIX system	11
9 Uninstalling WideStudio	12
9.1 Uninstalling on Windows	12
9.2 Unintalling on the UNIX system	12
9.3 Uninstalling rpm package on Linux system	12
B Getting started	13
1 What is the application builder?	13
2 The method of development of an application on WideStudio	15
3 How to start the application builder	16
3.1 How to start on MS-Windows	16
3.2 How to start on the UNIX system	16
4 How to exit the application builder	17
5 The name and function of the parts of the builder	18
5.1 Inspector,Source code inspector	18
5.2 Property editor	18
5.3 Event procedure editor	19
5.4 Window attribute editor	19
6 Development of a simple application "Hello"	20

7	Creating a project "Hello"	21
8	Creating an application window	22
9	Creating an event procedure.	23
10	Building of the application "Hello" and executing it.	25
C Application Window		26
1	What is application window?	26
2	How to create/save a new application window	27
3	How to see the list of the application windows	29
3.1	How to see the list of the application windows	29
3.2	How to display the application window to edit	29
4	How to delete the application window	30
5	How to rename the application window	31
6	How to save the application window as specified file name	33
7	How to open the file of an application window	34
D Objects		35
1	What is the GUI objects?	35
2	The Wiondows	37
3	The Forms	38
4	The Commands	39
5	The Drawings	40
6	How to place the instances on the application window	41
7	How to set the porperties	43
8	How to delete the instance	45
8.1	How to delete the instance	45
8.2	How to delete the application window	45
9	How to copy the instances	46
10	How to make the instance be an external variable	47

11	How to see a list of the instances	48
11.1	How to see a composition of the instaces of the application window	48
11.2	How to see the details of the children	48
12	How to rename the instances	50
13	How to define the instances as an array	51
14	How to use new classes of import libraries	52
15	How to see the class name of the placed object	54
E	Event Procedures	55
1	What is the event procedure?	55
1.1	The function for the event procedures	55
2	Triggers	56
3	How to create/setup/delete a event procedures	57
3.1	How to see the event procedures of the instance	57
3.2	How to create an event proceduree	57
3.3	How to setup the event procedure	58
3.4	How to delete the event procedure	58
4	How to create/edit the function	59
4.1	How to create a template file of the function	59
4.2	How to edit the function	59
4.3	How to appoint your favorite editor	59
4.4	How to use a function which is in a library.	60
F	Project	61
1	What is project?	61
2	The files in a project.	62
3	How to create new project/delete it/save as	63
3.1	How to see the name of project	63
3.2	How to create a new project	63
3.3	How to save project	63
3.4	How to rename project and save it	64
3.5	How to open a created project	64
4	How to set environment of project	65
5	How to add an application window to project	66
6	How to add a color	67
6.1	How to add a color	67

7	How to edit a color	67
8	How to delete a color	68
9	How to set up fonts	69
10	How to find applicatoin windows and instaces	70
G	Compiling/Building	71
1	How to build project	71
1.1	How to build project	71
1.2	How to appoint header path and library path	71
2	How to execute a compiled load module	73
3	How to appoint compiler options	74
3.1	How to appoint compile optionsr	74
4	How to appoint libraries to link	75
4.1	How to appoint libraries to link	75
5	Source file addition setting	76
5.1	Source file addition setting	76
6	How to appoint debugging mode	77
7	How to debug an application	78
8	How to use a trace debugger	80
H	Class Application Window	82
1	What is class application window?	82
2	How to create a new class application window	84
2.1	How to create a new class application window	84
2.2	How to appoint an icon and title string of the class	84
3	How to appoint class base for a new class	86
3.1	Default base class	86
3.2	How to appoint a base class for a new class	86
4	How to add a new property / edit / delete it.	88
4.1	How to display property setup dialog.	88
4.2	How to add a new property	88
4.3	How to edit a property	89
4.4	How to delete a property	89
4.5	How to create a new invisible property	89

5	How to delete/make invisible an existing property which is inherited	91
5.1	How to delete an existing property	91
5.2	How to make an existing property invisible	91
6	How to make child instances be member of the class	92
7	How to add/edit/delete triggers	93
7.1	How to display the trigger setup dialog	93
7.2	How to add triggers	93
7.3	How to delete added triggers	93
8	How to add/edit/delete a new user trigger.	94
8.1	How to display the user trigger setup dialog	94
8.2	How to add user triggers.	94
8.3	How to delete added user triggers.	94
9	How to create a class library	95
I	Stored Application Window	96
1	What is Stored Application Window?	96
2	How to make an stored application window	97
2.1	How to save an application window by stored type	97
2.2	How to add the stored application window to project	97
3	How to make a partial stored application window	98
J	Remote Instance	99
1	What is the Remote Instance	99
2	How to start up Agent	101
3	How to construct a remote instance server	102
3.1	How to get an appliation become a remote instance server	102
3.2	How to publisize an instance as a remote instance	102
3.3	How to start up an agent	102
4	What is wsagent	104
5	Before accessing remote instances	104
6	How to access remote instances	104

A Introduction

1 What is WideStudio?

WideStudio is the GUI based application development environment which is born purely in Japan run on multiplatforms such as Windows, Linux, FreeBSD, and Solaris. This is fully freely available and an "open source" that is its most standout characteristic. Everyone can get this and develop open and multi-platform applications quickly. It is sure that users will find this attractive. WideStudio has following characteristics:

- Open source and Licencing with X11/MIT
- C++ based development
- Possible to build plantform-independent window applications
- Using fully original class libraries which are dependent on no other libraries.
- Equipped with an application builder which can easily edit an window application
- Needed only a minimam-coding by an event-driven style
- Ensure a scalability by a function of building/importing an additional class libraries

Mature developers may have developed applications on multiplatforms such as Windows and X11. I too have developed various applications till now. Storing money, getting a palm-top computer, input the manchine language on it drawing by dots to make so called a game, which is a old time story. Since then, I used MS-DOS follows Windows 3.0, 3.1, UNIX/X11 Window system, Xt/Motif, Windows95/NT(Win32), MFC/ActiveX. At last, I don't want to bother anymore from incompatibility between platforms and have decided to make platform-independent.

Well, it is important, besides, I thought it be great if developers could build applications only by sticking with mouse on display and prepare an application builder, WideStudio the integrated IDE. Next figure describes composition of WideStudio and its files.



[WideStudio Composition]

2 What is C++ language?

WideStudio is written in C++. C++ is a program language enhanced the C language to handle object oriented programing. C language is the one of the most standard programing language which covers some larger software products because of its hardware accessing ability. C++ employs such wonderful potential from C language and is equipped with functions in order developers to build softwares more efficiently. While, the more it traces C++ format, the more it becomes the world diverged from one in C, which would be inclined to get novice developers or developers who are familiar with C difficult to understand. Using WideStudio, developers can write programs as if they did in C in order to make good use of grat C++ functionalities, and of cause, programs using classes are also easy to make.

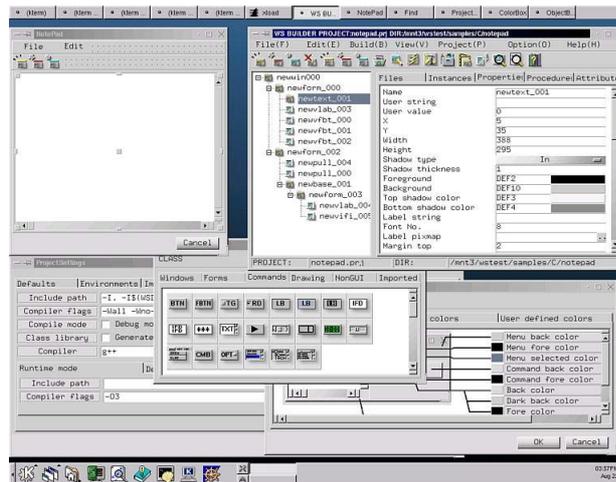
3 Through WideStudio development

Thinking twice, there are thousands of application builders like WideStudio. Then, why "WideStudio"? I often hear "I want to make some applications, but it's hard...", "I don't want to buy a shit application builder paying money.." and they cannot take up programing. WideStudio solves these problems. It's free, and so easy to use! And also I often hear in the world where open source is not familiar with, "What the hell is the open source?", "Why WideStudio is free?"

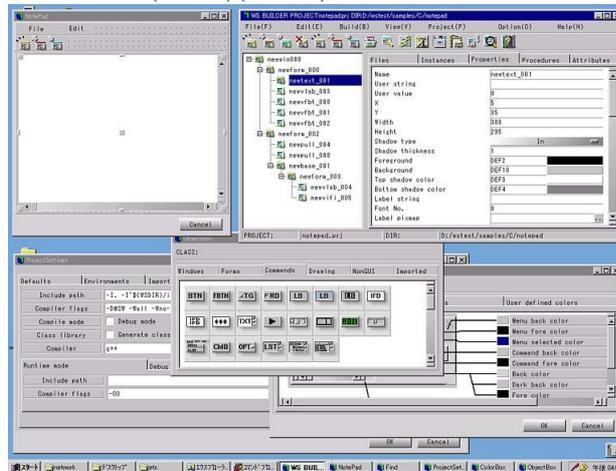
The object of developing WideStudio is to contribute it to the open source world that I have larned a lot from there. WideStudio was made in restricted time to see how efficiently it could be built/designed. I would appreciate if any part of this source would be helpful for developpers in developing/ designing softwares from now. From nuts to real applications, it will be much delight if everyone shares excitement in developing through engineering.

4 The multi-platform environment

Using WideStudio enables easily to develop multiplatform applications without aware of platforms in highly consistent environment. And the same applications developed by WideStudio can be run on Unix/Linux, Windows.

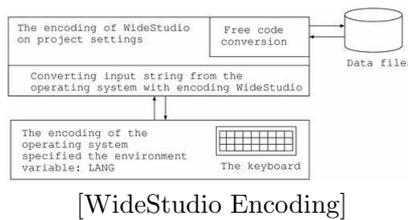


[WideStudio on Unix (Linux)(Above), WideStudio on Windows(Below)]



5 The multi encoding

Using WideStudio enables to develop applications without aware of different encoding among platforms. WideStudio has coding systems such as Unicode(UTF8), EUCJP, SJIS. WideStudio can deal with the encoding independent on the encoding on each platform so, for example, EUCJP applications built on Linux can also run on Windows With EUCJP codes without any modifications. Specifying LANG environment valuable, WideStudio decide a code system from keyboard input on operation system and tranforms automatically. Besides, it transforms into given code system in external data file I/O.



[Example, displays EUCJP code on Windows of English version]

Also, users can develop internationalized applications using unicode.



[Displaying each language by unicode]

WideStudio supports following encoding systems.

The encoding	
ISO8859_1	European
ISO8859_2	Eastern European
ISO8859_3	Turkish, Esperanto
ISO8859_4	Baltic
ISO8859_5	iso8859-5
ISO8859_6	Arabic
ISO8859_7	Greek
ISO8859_8	Hebrew
ISO8859_9	Turkish
ISO8859_10	Baltic2
ISO8859_13	ISO8859(13)
ISO8859_14	ISO8859(14)
ISO8859_15	European 2
UTF8	UNICODE UTF8
KOI8R	KOI8 Russian
EUCJP	EUC Japanese
SJIS	SJIS Japanese
EUCKR	EUC Korean
EUCCN	EUC Chinese
BIG5	BIG5 Chinese

6 The open source license

WideStudio is an open source. Everyone can access to the source code of WideStudio, and re-distribute it freely.

WideStudio can be used for business also with no charge, because WideStudio is X11/MIT license.

- Free re-distribution of the source code
- X11/MIT License
- Free commercial use
- No guarantee

7 Downloading of WideStudio

It is possible to download WideStudio from as following URL.

<http://www.widestudio.org> (original site)

<http://www.sourceforge.net/widestudio>

<http://www.vector.co.jp>

8 Building of WideStudio and installing it

8.1 Building and installing on the UNIX system

At first, download the source code and online manual from WideStudio home page, and unpack it as follows.

Notice: vX.X depends on the file version.

On linux operating system

```
cd /tmp
tar -zsvf ws-vX.X.tar.gz
cd /usr/local/ws
tar -zsvf ws-vX.X-doc.tar.gz
```

On SOLARIS operation system

```
cd /tmp
gzip -cd ws-vX.X.tar.gz | tar -moxvf -
cd /opt/ws
gzip -cd ws-vX.X-doc.tar.gz | tar -moxvf -
```

There are several targets on building of WideStudio.

They are Runtime libraries, Debugging libraries and so on. We usually use the runtime libraries, and use the libraries for debugging on debugging/development of the WideStudio application. So it is expected to build the libraries for debugging, If the user develop the WideStudio application.

Notice: "ws" is indicate the directory which is WideStudio installed in the examples.

Notice: Sometime it will occur some compile errors by a difference of machine environment. In such case, you can edit the file: ws/sys/config/mkfiles to configure the compiler flags, the include path, the libraries path and so on, to avoid compile errors.

In the case on Free BSD, use command "gmake" instead of command "make".

To build all

```
cd ws/src
./configure
make
```

To build the runtime libraries and the application builder

```
cd ws/src
./configure
make runtime
```

To build the runtime libraries for debugging.

```
cd ws/src
./configure
make debug
```

If finished building, please confirm whether there is the application builder (ws/bin/wsbuilder). If exists, it is success. Execute the following command by super user to install it.

```
cd ws/src
make install
```

8.2 Installing with the binary rpm package on Linux operating system

It is easy to install with the binary rpm package. It requires no building the source code.

Execute the following command by super user to install it.

Notice: vX.X depends on the file version.

```
rpm -i ws-runtime-vXXXX.i386.rpm
```

WideStudio will be installed into the directory /usr/local/ws. The rpm package of WideStudio requires some C++ runtime library rpm package, so it it occurs some error like [libstdc++-libc6.1-1.so.2 not found], please install it too.

8.3 installing on Windows operating system

At first,unpack the downloaded zip file, a directory "wsinst" is available. In order to install, execute the executable: "setup.exe" in the directory. The following wizard dialog will be indicated, then specify the install directory.



[The setup wizard dialog of WideStudio]

Next, specify your favorite Web browser.



[Settings of the Web browser]

Next, specify your favorite source code editor.



[Settings of the source code editor]

Next, it indicates the license agreement of WideStudio and GNU, and installing is done. Reboot your computer(This is important).

8.4 Setting of the environment variables on Windows

The installer of WideStudio sets the environment variables up, the user do not need to set them. It requires to reboot after installing WideStudio in order to reflect them.

8.5 Setting of the environment variables on the UNIX system

It requires to set the following environment variables up to make WideStudio ready after installing it.

WSDIR	The install directory of WideStudio
PATH	The command directory of WideStudio
LD_LIBRARY_PATH	The shared library directory of WideStudio

In the case of unsetting the environment variable "WSDIR", The application builder treat it as following value.

System	The default value
Linux	/usr/local/ws/
SunOS	/opt/ws/
The other UNIX	/usr/local/ws/

For example, In the case the following install directory is "/export/home/ws", Set the environment variables as follows.

(Note: xxxxx in the list shows the value before modification.)

An example of installed directory	/export/home/ws/
in csh	setenv WSDIR /export/home/ws setenv PATH xxxxxx:/export/home/ws/bin setenv LD_LIBRARY_PATH xxxxxx:/export/home/ws/lib
in sh	WSDIR="/export/home/ws";export WSDIR PATH="xxxxxx:/export/home/ws/bin";export PATH LD_LIBRARY_PATH="xxxxxx:/export/home/ws/lib" export LD_LIBRARY_PATH

Please add these settings to .cshrc(if use csh), .profile (if use sh), and reflect it by logging out or execute rehash command to update the values.

Please refer the manual of shell, if you needs the details of setting of shell environment variables.

8.6 The other files on the UNIX system

The files which are used by the application builder are as followings. These files are automatically created by the application builder in the home directory.

The file name	Description
.wsrsc	Contains the settings of look and feel, default colors.
.wsbuilderrc	Contains the specified editor, browser
.wsbuilder_defaults	Contains the information of the application builder

The command: wsreset resets these files in order to initialize the settings, if some files contains wrong value.

9 Uninstalling WideStudio

9.1 Uninstalling on Windows

The user can uninstall WideStudio by selecting it on the deleting application dialog of the control panel dialog. The user can uninstall to execute the following executable "unsetup.exe" too.

```
WideStudio/ws/bin/unsetup.exe
```

9.2 Unintalling on the UNIX system

In the case that the user built WideStudio by executing "make" and installed by "make install", It is possible to uninstall by "make uninstall". Change directory to the building directory of WideStudio, and execute "make uninstall" by super user.

```
make uninstall
```

9.3 Uninstalling rpm package on Linux system

In the case that the user installed by the binary rpm package of WideStudio, it is easy to uninstall it by "rpm" command. Execute the following command by super user.

```
rpm -e ws
```

B Getting started

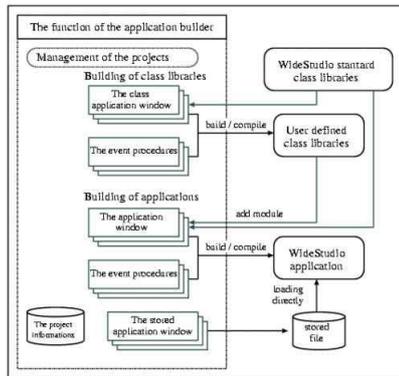
1 What is the application builder?

The application builder is very strong and very useable program, and draws out the maximum ability of WideStudio which is an integrated development environment for gui applications. Application developers are released from complicated implementation of gui applications by the application builder.

Specification of the application is as follows.

- Making and Managing the application windows
Making the application window with interactive operations, management of them.
- Implementation with the event procedures
Minimum coding of applications by an event driven system. Developers prepare the event procedure and set them to the gui objects.
- Properties of the objects
Developers can modify the properties of the objects with interactive operations.
- Informations of the application
Developers can refer the objects (the instances) and the event procesures, their properties,other informations by the application builder.
- Manament as project
Management of the files of the application window and files of the event procedures, the environment for development and building by project unit.
- Automatic building of the application
Generation of makefile ,c++ source codes of the application window, and compiling, linking them, and creates the executable.
- Executing and debugging of the application
Executing and debugging the built application.
- Supporting of the class application window
The class application window is built as new c++ class of WideStudio. The application builder generates a class library by integrating the clases, and makes it possible to use it from the other applications.
- Suppoting of the stored application window
The application builder stores the application window to file. It is possible that the application can load it directly, and share it. Treating objects as stored application window is effective for the applications which displays some maps,drawings and wiring diagrams.

The following shows the function of application builder, and structure of the files.



[Abstract of the application builder]

2 The method of development of an application on WideStudio

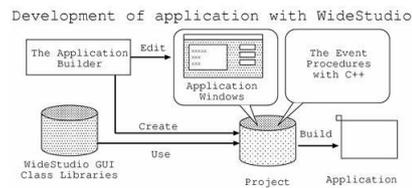
The method of development by WideStudio Application Builder is as followings.

- Creating a project

The developer creates a project for an application at first. A project is an unit of application development. For detail,see the Chapter 6.
- Creating the application windows

The developer can develop application windows visually by WideStudio Application Builder. For detail,see the Chapter 3.
- Creating the event procedures

The event procedure is executed by firing the event. The developer can implement the procedures which do some advanced processing like complicated painting and data processing by C/C++ language easily, and can choice the event to execute it. For detail,see the Chapter 5.

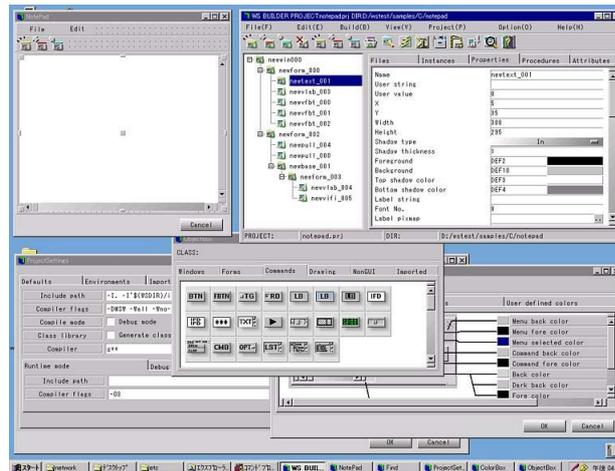


[The method of development of an application]

3 How to start the application builder

3.1 How to start on MS-Windows

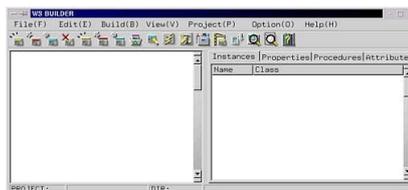
The user can start the application builder by clicking wsbuilder icon in the Windows start menu.



[The application builder on MS-Windows]

3.2 How to start on the UNIX system

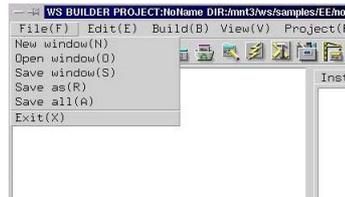
The user can start the application builder with wsbuilder command on the X-Window system. It requires the c++ compiler, debugger, source code editor to build applications.



[An initial view of the application builder]

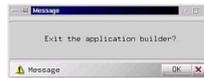
4 How to exit the application builder

Select [Exit] of the menu:[File] to exit the application builder, or push a button: [X] under the right.



[Exiting of the application builder]

The following dialog pops up, and then push its button:[OK]. if there are unsaved application window or project, the question dialog pops up to confirm saving them.



[The exiting dialog]

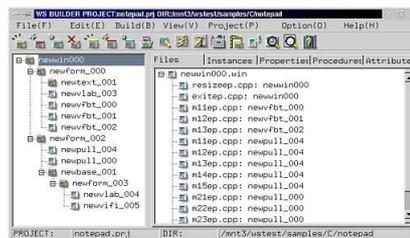
5 The name and function of the parts of the builder

With WideStudio Application Builder, the application developer can develop an application visually. The builder has as following parts/windows.

- Inspector
- Source code inspector
- Property editor
- Event procedure editor
- Window attribute editor

5.1 Inspector,Source code inspector

By Inspector, the developer can confirm the structure of the instances on the application window easily. In addition by Source code inspector, the developer can confirm which instance has the source codes of the event procedures.

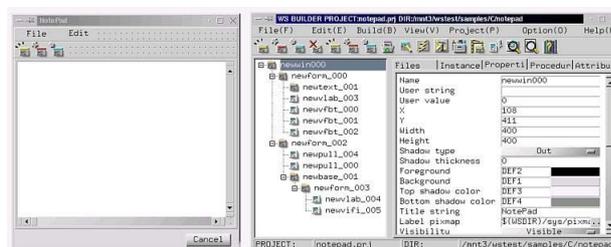


[The structure of the instances(left side) and source codes(right side)]

By clicking the instance in Inspector, it becomes selected status. In the case it has [+]/[-] mark,it has child instances on it,and by double clicking,it open the instance and indicates the children on the instance tree.

5.2 Property editor

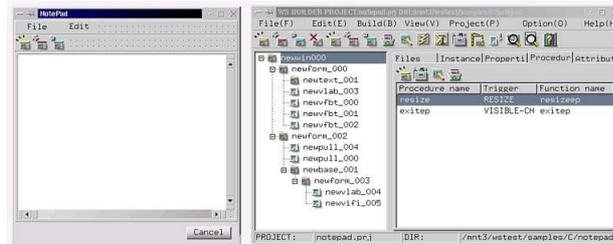
The instances of WideStudio has the internal variable as the properties which defines the geometry, colors, shapes,types,and so on. And, these properties can be changed visually by Property editor of the builder.



[Changing values of the property by Property editor]

5.3 Event procedure editor

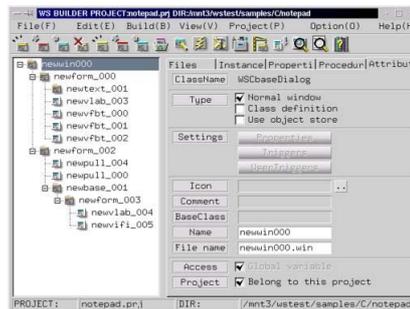
It requires the event driven programming on WideStudio. The developer puts some event procedures to the instances with some trigger. If the event procedure is on the push button instance with ACTIVATE trigger, by clicking the instance, the trigger will be fired and it will execute the procedure.



[Editing the event procedures by Event procedure editor(right side)]

5.4 Window attribute editor

The instances have some attributes like as instance definition type, and so on. The developer can edit it by Window attribute editor.



[Editing the attributes by Window attribute editor(right side)]

6 Development of a simple application "Hello"

Let's develop a simple application "Hello" with WideStudio Application Builder. The application has a push button, and by pushing it, it indicates "Hello!" on the button. The development of a simple application "Hello" supports understanding the development flow of WideStudio.

The application indicates "Hello!" by clicking the push button.



[A simple application "Hello"]

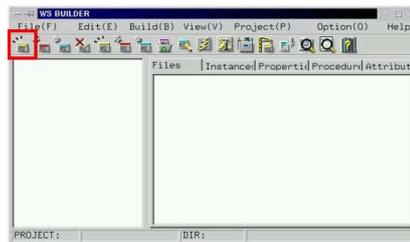
It makes the application "Hello" by the flow as follows.

- Creating a project for "Hello".
- Creating an application window for "Hello".
- Creating an event procedure to indicate "Hello!".
- Building an application "Hello".

7 Creating a project "Hello"

Let's create a project which is for the application "Hello". To indicate the wizard dialog, select [New project] of [Project] menu of the application builder, or click the following icon.

To the wizard dialog, input the project name (the application name) "hello" and directory name, and choice [Normal application] type.



[Creating a project for the application "Hello"]

8 Creating an application window

On WideStudio, They say "Appcagation Window" about the window which is developed by user. Now, create an application window for the application "Hello". Select [New window] of the [File] menu and choice [Normal window] of the window type of the wizard dialog. and input the application name "newwin000" to it and create window with no template.



[The wizard dialog for creating application window]

The ObjectBox dialog is used to put GUI instances into the application window. In order to indicate it, select [ObjectBox] of [View] menu. Then a dialog will appears as following picture.



[The ObjectBox dialog]

In order to drag and drop a push button instance to the window, click the [Commands] index tab of the dialog, and drag the icon [BTN] (WSCvbtn/push button class) and drop it into the window as following picture.

It is possible to edit the properties of the instance with WideStudio Application Builder. So let's change the figure of the push button instance. Select the instance, and click the [Properties] tab on the application builder, and set properties as follows.

Name: newvbtn_000
 X: 10
 Y: 10
 Width: 200
 Height: 30
 Label string: push!

It becomes as following picture.



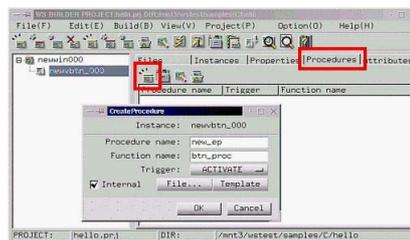
[The application window "Hello" under constructing]

9 Creating an event procedure.

The instance fires the various kind of event on executing. On WideStudio,It is possible to put some event procedure on the instance which procedure will be executed with the event fired. For example, let's create an event procedure which has the following function.

Indicating "Hello!" on the button by clicking it on the window.

Set an event procedure to the instance so that it execute the procedure by clicking the instance: newvbtn_000. At first,select the instance: newvbtn_000 by mouse pointer, and and select the [Procedures] index tab of the inspector,and click the following icon to create an event procedure.



[Creating an event procedure]

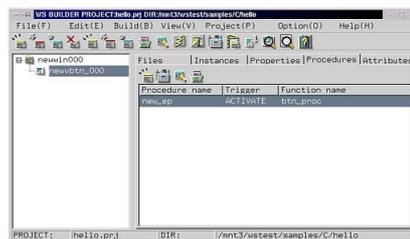
Input the procedure name,the C/C++ function name,and so on,to the dialog as follows.

Procedure name: new_ep

Function name: btn_proc

Trigger: ACTIVATE

The procedure name is used for identification of the event procedure. it is possible to set any name for it. The function name is a C/C++ language function name which function the application developer write some program to do something. The last one is the trigger, The ACTIVATE trigger will be fired by clicking the button normally. It is possible to create a file of the template source code for the event procedure.



[A event procedure.]

It indicates an editor by double clicking the item of the list as following event procedure. It is possible to appoint it which is your favorite. Select [Project settings] of the [Project] menu and input your favorite

editor into the project settings dialog. The default editor is vi on the UNIX system and notepad.exe on the Window system.

```

#include <WScom.h>
#include <WSCfunctionList.h>
#include <WSCbase.h>
//-----
//Function for the event procedure
//-----
void btn\_ep(WSCbase* object){
    object->setProperty(WSNlabelString,"Hello!"); //A
}
static WSCfunctionRegister op("btn\_ep1", (void*)btn\_ep1);

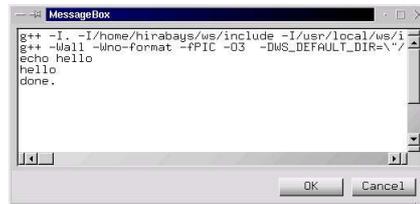
```

The point object is the instance which has this event procedure. At //A, set a string "Hello!" to the property: WSNlabelString of the instance Then, the instance indicates "Hello!".

10 Building of the application "Hello" and executing it.

It is better to save the created data of this project, at first. Select [Save project] of the [Project] menu to do it.

Then, the operation to do next is to build the application. Select [Build all] of the [Build] menu, and the building dialog will be appeared in order to confirm whether there are compiling errors. If no error occurs, it is completion and the executable "hello" is available in the current directory. It is possible to execute it by selecting [Execute] of the [Build] menu.



```
g++ -I. -I/home/hirabays/ws/include -I/usr/local/ws/i
g++ -Wall -Wno-format -fPIC -O3 -DWS_DEFAULT_DIR=~/
echo hello
hello
done.
```

[Building the application "Hello"]

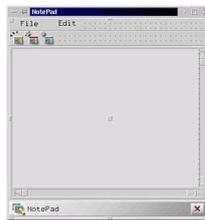
C Application Window

1 What is application window?

"application window" is a window which is created by the application builder. The application builder manages and saves it each application window unit. It is a unit of the top level window. There are the following type about application window.

- Normal application window
Usually you create the application with the normal application window. we call it simply "application window".
- Class application window
You can create a new c++ class by integrating some classes and some event rocedures all together which are often used by application. we call it "class application window". This function promotes efficiency of development.
- Stored application window
The application window can be stored to a object file, and the application can loads it and generate instances directly. You can replace it dinamically,share it between several applications and processes.

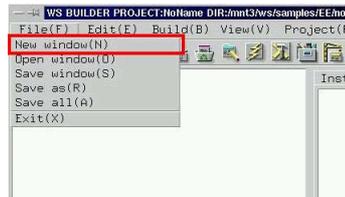
we call it "stored application window".



[An application window under construction...]

2 How to create/save a new application window

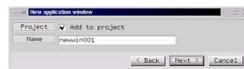
You can create a new application window with [New window] of the [File] menu. Then a wizard dialog for new application window pops up, you can appoint the type ,name of the application window, and push the [Finish] button to create it.



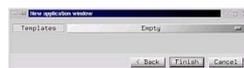
[Creating a new application window]



[The application window wizard (choice of type)]



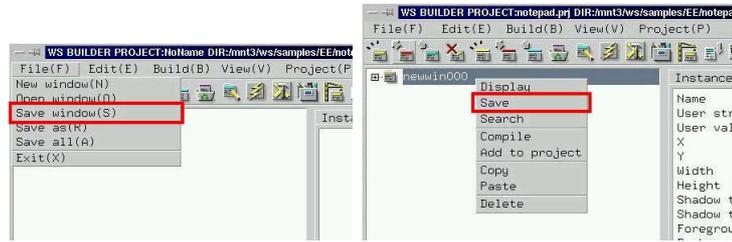
[The application window wizard (appointment of name)]



[The application window wizard (appointment of the template)]

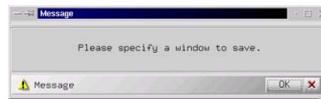
To save the application window, please select it by the inspector, and select [Save window] of the [File] menu. You can select [Save] of the popup menu too, which pops up by clicking the right button of the mouse pointer on the instance of the inspector,

The saved file is [window name].win.



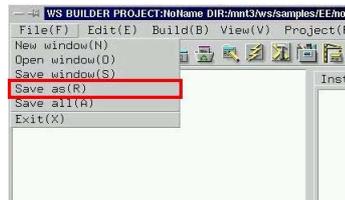
[Saving the application window]

A warning dialog pops up when you do not choose an application window to save. you may select [Save all] of the [File] menu, if you want to save all of the application window.



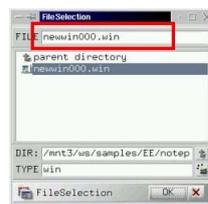
[A warning dialog]

You can rename and save the application window by selecting [Rename] of the [File] menu.



[Renaming the application window]

Please input a new file name into where the following shows.



[Input of a new file name]

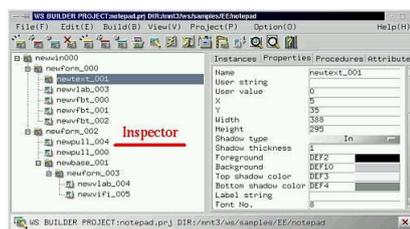
3 How to see the list of the application windows

3.1 How to see the list of the application windows

The inspector of the application window shows the list of the application windows and controls it.

It shows a tree style view to recognize the constitution of the instances easily which are placed on the application window.

The top instance of the tree is an application window. For example, An instance: newwin000 in the following figure is an application window.

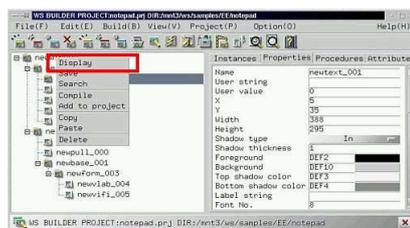


[Shows the tree of instances of an application window]

3.2 How to display the application window to edit

To display the application window to edit, please select it by the inspector, and select [Display] of the [Edit] menu. Or you can select [Display] of the popup menu which pops up by clicking the right button of the mouse pointer on the inspector.

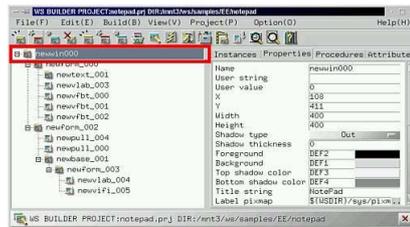
if you want to invisible it, select [Dismiss] of the popup menu of the instance on the inspector.



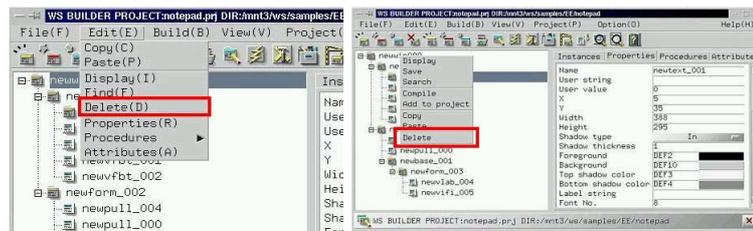
[Displays /Dismisses the application]

4 How to delete the application window

To delete the application window, please select it by the inspector like the following figure, and select [Delete] of the [Edit] menu or [Delete] of the popup menu which pops up by clicking the right button of the mouse pointer on inspector.



[Choosing the application window]

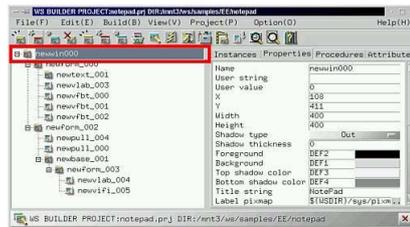


[Deleting the application window]

The question dialog pops up once, you can delete the instance by clicking the [OK] button.

5 How to rename the application window

There are two ways to rename the application window. The first one is to change the property: WSNname, another is to change with the input field in the section:[Attributes] of the inspector. Select the instance to rename in the inspector.

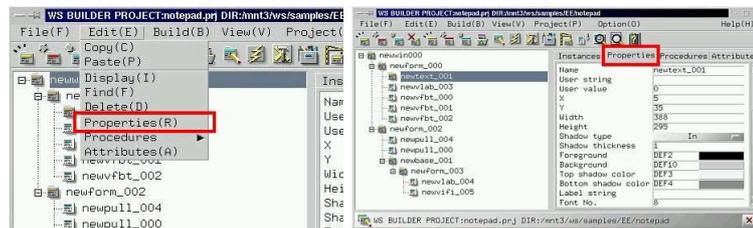


[Choosing an application window to rename]

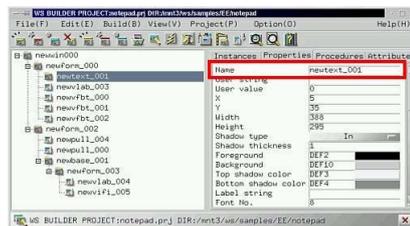
On the first way, select [Properties] of the [Edit] menu of the application builder, or push the [Properties] tab of the inspector.

Then the properties are shown, and you can rename the instance of the application window by the property: WSNname

(Notice) the name of an application window is equal to the name of the instance which is a top window.



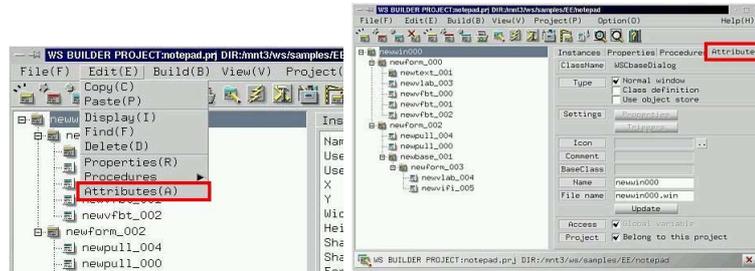
[The properties]



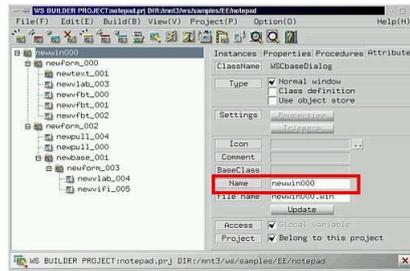
[Editing the name: WSNname property]

by [Attributes] window of the inspector, choose [Attributes] of the [Edit] menu, or the [Attributes] tab of the inspector.

Then the attribute window is shown and you can input a new name into the [Name] field and press the [Update] button. (Notice) It is equal to the property: WSNname of the instance.



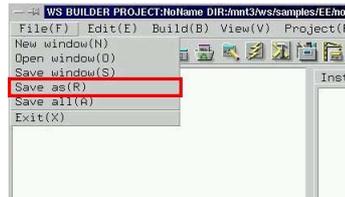
[The attributes]



[Changing the name]

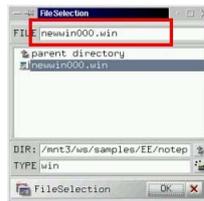
6 How to save the application window as specified file name

Usually, the file name of the application window is [name].win. you can save it as the other file name by [Save as] of the [File] menu.



[Save as]

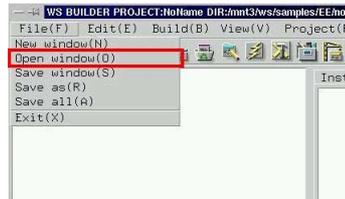
The following figure indicates where you input a new file name.



[Input a new file name]

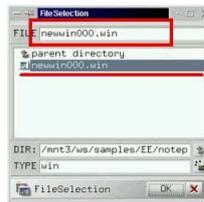
7 How to open the file of an application window

Select [Open window] of the [File] menu to load an application window from a file.



[Open an application window]

The following shows the file selection dialog to select a file of an application window.



[Selecting the file]

You can input directly, or select by double clicking of the mouse pointer.

D Objects

1 What is the GUI objects?

the GUI objects like the window ,push button are made by c++ class, and the has the followings function to customize easily.

- Property
The GUI objects has properties which likes a kind of member variables, and the depeloper handle them with the unified insterface. There are properties like the color, width, shape, value, string,and so on.
- Event Procedure
The developer set the event procedures to the instance of the GUI objects. The procedure has a trigger,and is executed on its trigger fired. The procedure handles it for the instance so that you can add it to the instances various kind of function which the objects do not have by procedure. Therefore, it makes you freed from creating many new classes which are unnesessary.

The following shows kinds of the GUI object which has the child management function.

Kind	Class Name	Description
Windows	WSCwindow	A top window as base of the application window
	WSCbaseDialog	A base of the dialog.
Forms	WSCform	An area which has independed window, and manages children.
	WSCscrForm	A scrolled area.
	WSCsform	An area which has several sub-areas which are resized freely by separators.
	WSCindexForm	An area which has tabs to switch the contents

The following shows kinds of the GUI object which does not have the child management function.

Kind	Class Name	Description
Labels/Buttons	WSCvbtn	A push button.
	WSCvtoggle	A toggle(radio) button.
	WSCvlabel	A text label which has a border.
Controls	WSCpulldownMenu	An pulldown menu.
	WSCoption	An option menu which choose a value.
	WSCpopupMenu	A popup menu.
	WSCvslider	A slider which adjusts a value.
	WSCvclock	A clock.
	WSClist	A list of strings.
	WSCvifield	A text input field.
	WSCvmifield	A multi-line text input field.
WSCtextField	A scrolled text input field.	
WSCcomboBox	A input field and menu to choose a string.	
Primitives	WSCvarc	An arc or circle.
	WSCvrect	A rectangle.
	WSCvline	A line.
	WSCvpoly	A poligon.

2 The Wiondows

The Window parts can be categorized roughly into two parts: Windows for general use and Dialog windows for prompting user to input something.

Usually, WSCmainWindows class is used for a top window, but you can set any class name, such as Dialog, as a top window to set the default setting on a project setting window.

WSCmainWindow

Used for a general top shell window. The instance of this class is created when new application window is created by window in default.

WSCdialog

Base window for a Dialog. Dialog equips "OK" and "NO" buttons beforehand, you can get these buttons shown and get this window focused in need.

WSCmessageDialog

Dialog for displaying messages. Show messages to get users noticed.

WSCfileSelect

Dialog for file selection. Use this to get users choice a file or a directory.

WSCwizardDialog

Dialog with switching screen function by the "¡ Back" "Next ¿" buttons.

3 The Forms

The Forms manages window region separately to the parent window placing other objects on its Form. Objects on a Form move along as the Form moves. Also, when a Form is visualized, the objects on it too.

WSCform

Used for a general Form. This Form provides dependent coordinates from the parent window, and raises resize event. When a resize event is raised, an event procedure for that event enables to complex replacing which is difficult to do for users by only setting anchor properties.

WSCscrForm

Form with scrolling function. Useful for handling larger area than an actual window size. Use Scroll bar to scroll the area. You can set various properties such as the size of the area, the scrolling unit and so on.

WSCsform

Form with separator to separate region. This separator has a mouse scaling to decide the size of the area.

WSCindexForm

Form with indexing tab. Selecting an index to show corresponding area. This form appears as if it has multiple layers.

4 The Commands

Command controls are: chiefly to use for displaying, to raise event by users using mouse(Push buttons), to select a value(Selecting value), to select an action out of menu(Pull-down menu), Input/edit strings by keyboard(Pull-down menu).

DISPLAYING

WSCvlabel(Strings,Pictures),WSCvclock(Clock),
WSCvmeter(Meter)

PUSHING BUTTON

WSCvbtn,WSCvfbtn
Click on mouse raises ACTIVATE event.

INPUT VIA KEYBOARD

WSCvifield,WSCvmifield,WSCtextField
All of them used for input texts.

SELECTING VALUE

WSCvradio,WSCoption,WSCcomboBox,
WSClist,WSCvslider,
Used for selecting a value.

PULL-DOWN MENUS

WSCpulldownMenu,WSCpopupMenu,
An event procedure on the menu item activates when the item is selected.

5 The Drawings

Drawing controls indicates various kinds of graphics.

Circle,Oval,Arc,Filled Circle
WSCvarc

Line,Poly-line
WSCvline

Rectangle,Filled rectangle
WSCvrect

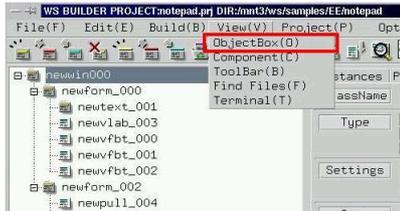
Polygon,Filled polygon
WSCvpoly

Free graphics
WSCvdrawingArea

Graph
WSCvbarGraph(the bar graph),WSCvlineGraph(the line graph),
WSCvgraphScale(the scale of the graph), WSCvgraphMatrix(the lattice of the graph)

6 How to place the instances on the application window

The dialog for creating the instance is the object box dialog. Select [ObjectBox] of the [View] menu of the application builder to popup.



[To popup the object box dialog]

The following figure shows the object box dialog.

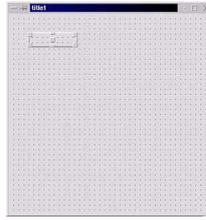


[A view of the object box dialog]

If you move the mouse pointer into the icons of the object box dialog, It pops up a balloon help which shows its class name, and some description. You can drag and drop the icon to the instance on the application window which is displayed. See the section: [How to see the list of the application windows.]. After displaying the application window, drag the icon and drop it on the instance to place.

The following shows kinds of the objects.

Kind	Description
Windows	Top windows or dialogs which have a border of the window manager.
Forms	Areas which have window resource, and manage children.
Commands	Objects which have various function, and do not have window resource.
Imported	Objects which is imported by the external libraries.



[A new push button on the application window.]

We call the object which manages the child instances "Manager". The following shows manager objects

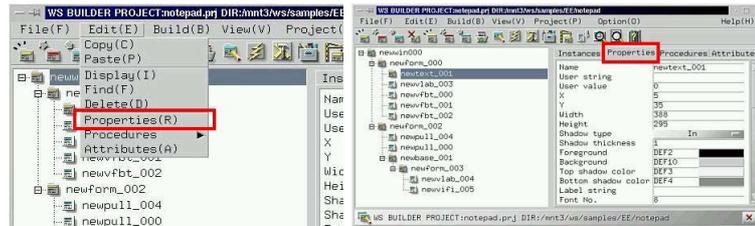
Manager	Class Name	Description
Windows	WSCwindow	A top window which is base of the application window.
	WSCbaseDialog	A dialog.
Forms	WSCform	An area which has window resources.
	WSCscrForm	A scrolled area.
	WSCsform	An area which has several sub-areas and separators.
	WSCindexForm	A area which has some index tabs

The instances of manager objects shows dot matrixed in editing mode on the application builder.

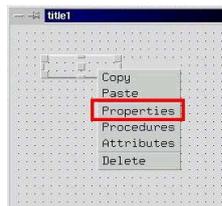
7 How to set the properties

The GUI objects have some properties which is a kind of internal variables. You can customize the instance about shapes, colors, actions., by its properties.

The property window of the inspector is for the setting of properties. Select [Properties] of the [Edit] menu of the builder, or [Properties] tab of the inspector to display it.

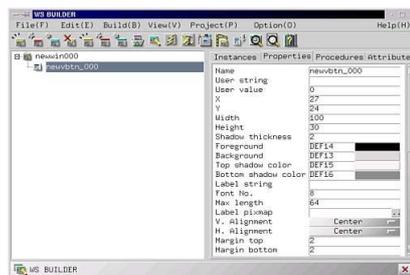


[The property window from the menus]



[The property window from the instance]

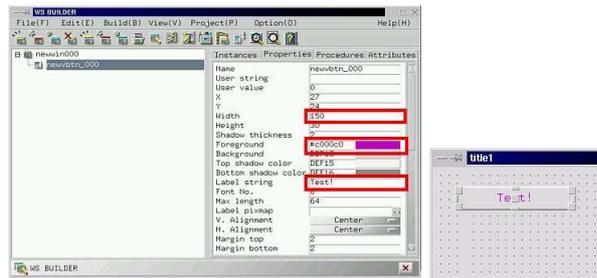
Then, the following window appears. It shows the properties of WSCvbtn (push button).



[A view of the property window (right of the inspector)]

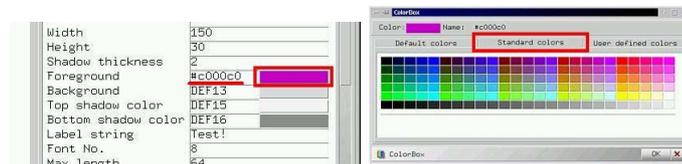
You can change the values of the properties and reflects by return key.

The following shows a change of value at underline, you can see the instance has changed.



[A change of value with the property window]

You can specify a hexadecimal RGB value, or press color button to display the color selection dialog, when the property is a color property.



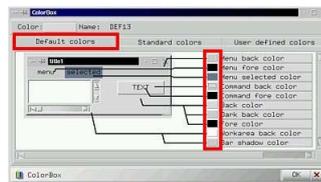
[Settings of the color property]

The color value is # + RR + GG + BB by hexadecimal. For example, White is #ffffff, black is #000000, Red is #ff0000, Blue is #0000ff, Green is #00ff00.

Valid value of the color value is as follows.

#000000 - #ffffff

You can also specify system default colors with the color selection dialog. You have better use these colors on development for Windows system.

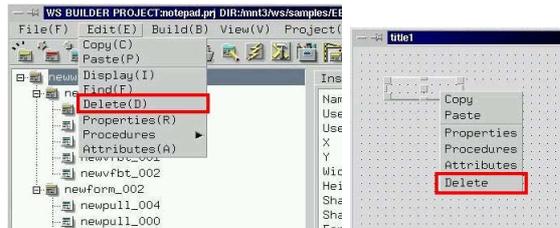


[Use the system default colors]

8 How to delete the instance

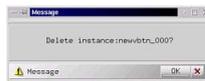
8.1 How to delete the instance

Select the instance which you want to delete at first, and select [Delete] of the [Edit] menu or [Delete] of the right button popup menu.



[Deleting the instance]

The following dialog pops up, then push the [OK] button. It deletes the child instances which places the instance too.

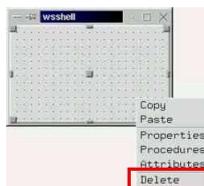


[Confirmation of deleting instance]

8.2 How to delete the application window

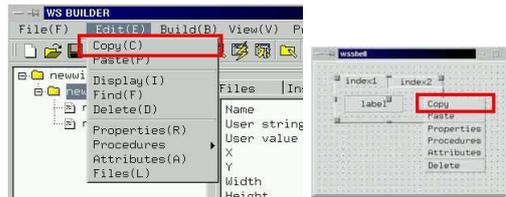
Instead of the way that is described in the Chapter 1, there is another way to delete the instance by the right button popup menu on the application window same as the usual instance.

The following figure shows to delete it by the popup menu.



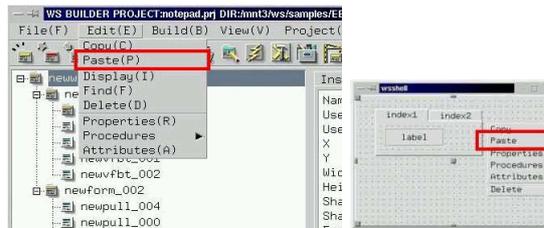
9 How to copy the instances

At first, select the instance to copy with the inspector or by clicking directly, and select [Copy] of the [Edit] menu or [Copy] of the right button popup menu on the instance.



[Selecting the instance to copy]

Select the instance to paste with the inspector or by clicking directly, next. then you can paste the instances on it, with [Paste] of the [Edit] menu or [Paste] of the right button popup menu.



[Selecting the instance to paste]

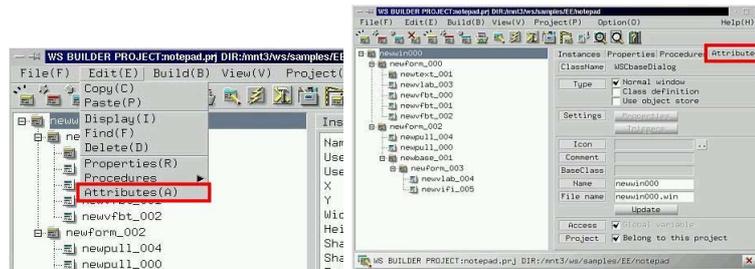
The following shows that it copies the instances.



[Copied instances]

10 How to make the instance be an external variable

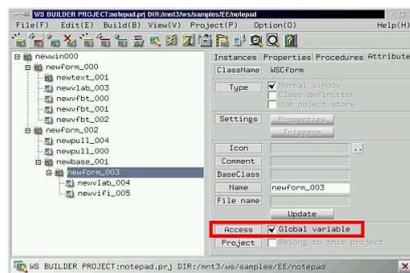
You can access the instance from c++ program like an event procedure as external variable, if you make it be an external variable. At first,select [Attributes] of the [Edit] menu,or the [Attributes] tab of the inspector.



[Attributes of the instance]

Check the radio button:[Access] and check it: Global variable.

But,the application windows and the instances which defined as array are always global/external variables.

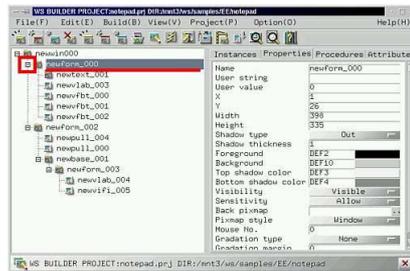


[Make the instance a global/external variable]

11 How to see a list of the instances

11.1 How to see a composition of the instances of the application window

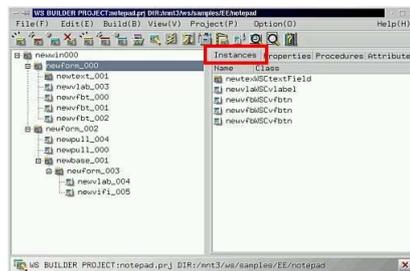
You can see a tree which shows the composition of the instances by clicking where the following figure indicates, or by double clicking the instance name. It shows the child instance of the clicked instance, and close them once again.



[A composition of the instances]

11.2 How to see the details of the children

You can see the details of the children by clicking [Instances] tab of the inspector.

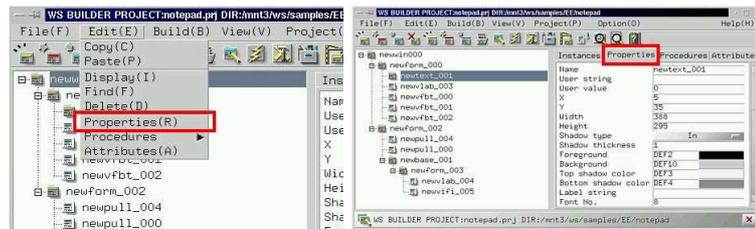


[The details of the children]

You can copy the children by multi-selection with the detail list. Select the instances as follows, and execute [Copy] and [Paste] of the [Edit] menu.

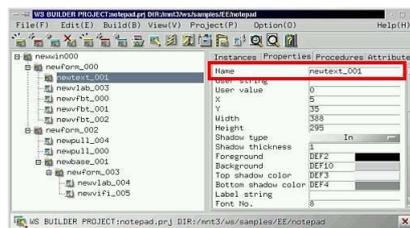
12 How to rename the instances

The application builder name the instances when they are placed on the application window. You can change the name with the property settings of the instance. Select [Properties] of the [Edit] menu or [Properties] tab of the inspector.



[The properties menu]

You can change the instance name with WSNname property. In addition, it reflects the application window name, When the instance is the top window of the application window. See the section:[How to rename the application window].



[WSNname property of the instance]

13 How to define the instances as an array

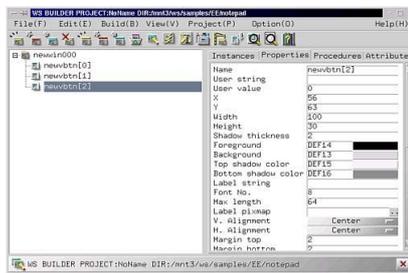
You can make the instances be an array by giving an array name. See the section:[How to rename the instances] to rename them.

Specify the array name as follows.

```
arrayname[No]
```

”arrayname” is given by alphabet and number which is used for an array name, ”No” is number which begins from 0.

You must make the instances be an array which class type is same one, and they must be on the same application window. So you can not make the application windows be an array.



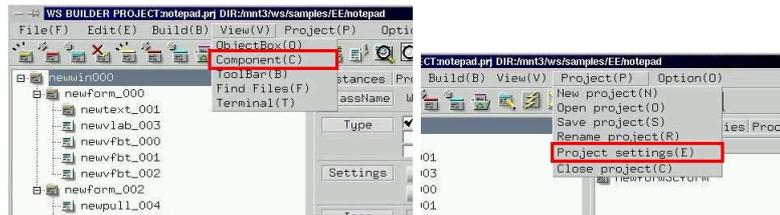
[The arrayed instance.]

See the section:[How to access to the arrayed instance] of the programming guide’s guide.

14 How to use new classes of import libraries

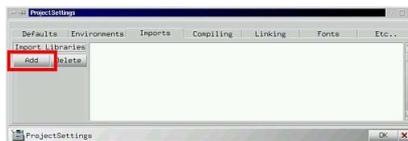
You can use new classes of import libraries on the application builder. specify import libraries with project setting dialog:[Imports] section which pops up by [Component] of the [View] menu or [Project settings] of the [Project] menu.

If you add them,you should specify the linking libraries on the section [Linking] of the dialog, too. See section: [How to specify libraries for linkage] for detail.

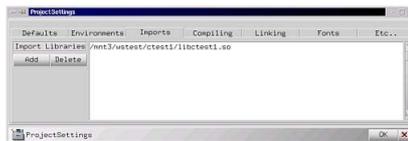


[View of the project setting dialog]

Push the [Add] button to add a import library, and specify it on the file selection dialog which is popped up. The import library is a dynamic link library for WideStudio.



[Importing a library]

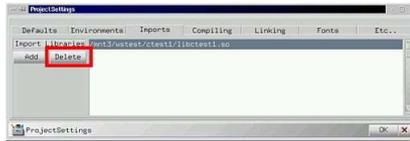


[a library imported]



[A class which is in the imported library]

If you want change the imported library, make it selected on the project setting dialog that following figure shows. And press the [Delete] button to delete it,then add a new library you want.



[Deleting the imported library]

15 How to see the class name of the placed object

Selecting [Properties] from [Edit] menu to see class name of the placed object. It displays class name in the class name form.

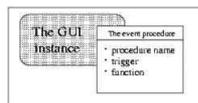
E Event Procedures

1 What is the event procedure?

The event procedure is a procedure which you can code with usual knowledge of C++ language. It is used to do something on the event fired, and to realize a complicated function which is too difficult to realize with only object.

It has a trigger and a client instance, and is executed when the trigger is fired on the client. See the programmer's guide. The following shows what the event procedure has.

- A name of the procedure
The instance has many event procedures,so they have their name to identify.
- A c++ function
The event procedure has a function which coded with c++ language. Tt executes the function,if the event is fired.
- A trigger
The event procedure waits for the event on the client instance. If the event is fired,it executes its function.



[The event procedure]

1.1 The function for the event procedures

The function has a parameter which passes the client object pointer.

The following function is a template that the application generated.

```

#include <WScom.h>
#include <WSCfunctionList.h>
#include <WSCbase.h>

//-----
//Function for the event procedure
//-----
void sample(WSCbase* object){
    //do something...
}
static WSCfunctionRegister op("sample",(void*)sample);
  
```

2 Triggers

There is 4 kinds of triggers as follows.

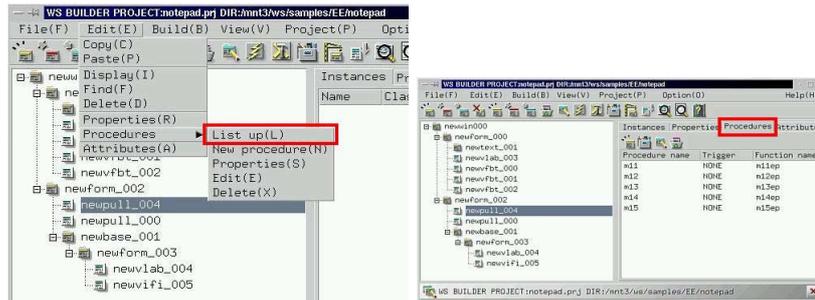
- Change of the state of the instance
It is fired by changing a state of the instance.
- The events of the mouse pointer
It is fired by the mouse pointer clicked, moved, ...
- The events of the keyboard
It is fired by the keyboard.
- The other

Change of the state	Description
WSEV_INITIALIZE	When the instance is initialized.
WSEV_DELETE	When the instance is released.
WSEV_ACTIVATE	When the instance is activated.
WSEV_VALUE_CH	When the value of the instance changes.
WSEV_VISIBLE_CH	When the visibility of the instance changes.
WSEV_PARENT_VISIBLE_CH	When the visibility of the parent instance changes.
WSEV_EXPOSE	When the instance is exposed.
WSEV_RESIZE	When the instance is resized.
WSEV_SENSITIVE_CH	When the sensitivity of the instance changes.
WSEV_PARENT_SENSITIVE_CH	When the sensitivity of the parent instance changes.
About the mouse pointer	Description
WSEV_MOUSE_IN	When the mouse pointer comes into the area.
WSEV_MOUSE_OUT	When the mouse pointer goes away out of the area.
WSEV_MOUSE_PRESS	When the mouse pointer is pressed.
WSEV_MOUSE_RELEASE	When the mouse pointer is released.
WSEV_MOUSE_MOVE	When the mouse pointer is moved on the area
About the keyboard	Description
WSEV_FOCUS_CH	When the focus changes.
WSEV_KEY_PRESS	When the key is pressed.
WSEV_KEY_RELEASE	When the key is released.
WSEV_KEY_HOOK	Hooking of the key event before despatching.
The other	Description
WSEV_NONE	None.

3 How to create/setup/delete a event procedures

3.1 How to see the event procedures of the instance

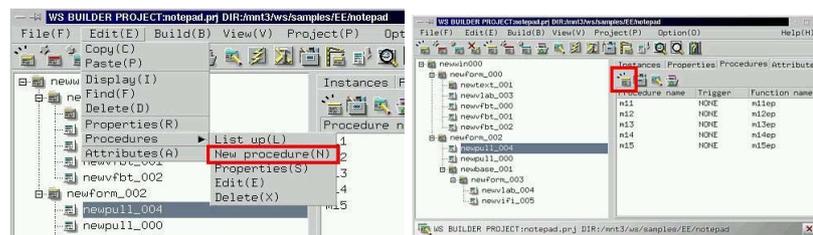
It is the procedure window of the inspector which is used for creating the event proceduses. Select [List up]/[Procedures] of the [Edit] menu or the [Procedures] tab of the inspector to display the procedure window.



[A view of the procedure window]

3.2 How to create an event procedure

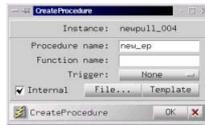
Select the target instance for event procedure on the inspector, and select [New procedure]/[Procedures] of the [Edit] menu, or the following icon.



[Creating an event procedure]

Input a name of procedure, trigger, function into the procedure setting window. Press [Template] button if you want a template code of a function for the event procedure.

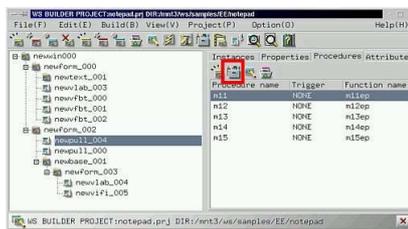
- Procedure name:
Input the procedure name which is used for identification of the event procedure.
- Function name:
Input a c++ function for the event procedure.
- Trigger:
Choose the trigger for executing.



[The settings of a new event procedure]

3.3 How to setup the event procedure

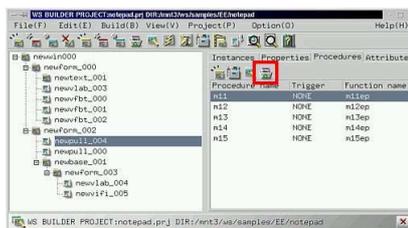
Select the event procedure like following picture, and select [Properties]/[Procedures] of the [Edit] menu or click the following icon to change the properties of the event procedure.



[The settings of the event procedure]

3.4 How to delete the event procedure

Select the event procedure to delete, and select [Delete]/[Procedures] of the [Edit] menu or click the following icon.

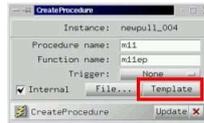


[Deleting the event procedure]

4 How to create/edit the function

4.1 How to create a template file of the function

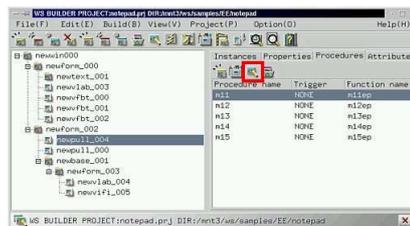
You can create a template file of the function by clicking the [Template] button after appointing a function name.



[Generating of a template]

4.2 How to edit the function

Click the following icon, or double click the procedure name, Then, the text editor is activated, and you can edit the function.



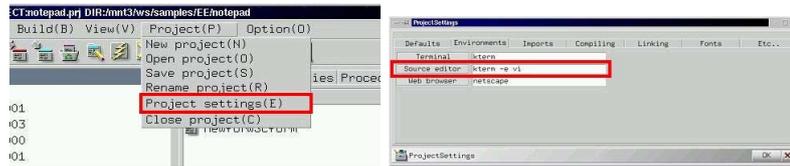
[Editing of function]



[A view of editor which stood up]

4.3 How to appoint your favorite editor

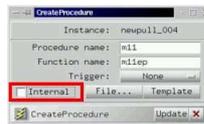
You can appoint your favorite editor with the environment settings window. Select [Project settings] of the [Project] menu, and select the [Environments] tab. Please input command name of your editor and arguments.



[Representation of the environment settings window]

4.4 How to use a function which is in a library.

You can use a compiled function which is in a library for the event procedure. Do not select the item of using source as follows. In this case, the application builder links the library, but does not compile and link object file of the function.



[Use a function of a library]

You need to specify a library which includes object file of the function in the linker options window. See the section:[How to build the project].

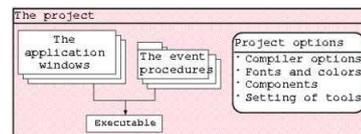
F Project

1 What is project?

A project is unit of administration of the application which developers build with the application builder. They define a project for an application when they build it.

There are many windows and many event procedures in the application. The developers need not to manage these source program one by one, Project does as follows.

- Administration of compilation of source program
Project compiles the managed application windows, event procedures, make up the application. There is a function that compiling updated source programs and rebuilding the load module.
- Administration of application windows in project
You can add application windows to project. The application builder generates the source programs of them, and compiles them automatically.
- Administration of event procedures in the application window
Project manages event procedure in application window too. They are compiled automatically, and become one part of the application.
- Administration of the working environment settings
Project contains the settings such as default of colors or dimensions, path of the include files/libraries for compiler options.



[Project]

2 The files in a project.

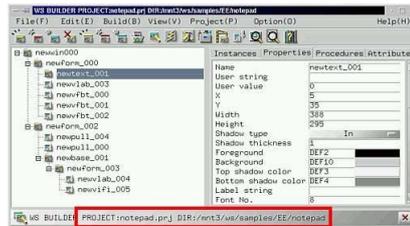
In developing an application with WideStudio Application Builder, Various kinds of file will be made by it. For example, there is a project "project1", and it has an application window "newwin000", the following files are created by the builder.

FILE NAME	DESCRIPTION
project1.prj	Contains the settings of project.
project1.cpp	A source code of the project.
project1.wns	A list of the application window.
project1.col	A list of the user defined colors.
newwin000.wiun	Data of the application window "newwin000"
newwin000.cpp	A source code of the application window "newwin000"

3 How to create new project/delete it/save as

3.1 How to see the name of project

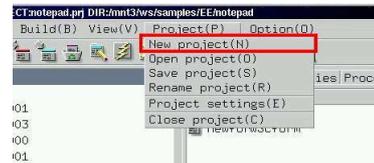
Look at the following figure to confirm the name of current project. If it is blanks,it means that none project is opened.



[Project name]

3.2 How to create a new project

Select [New project] of the [Project] menu,or click the following icon to create a new project.



[Creating a new project]

Then, a wizard dialog for new project is indicated, input a project name, specify the type.



[Wizard for new project]

3.3 How to save project

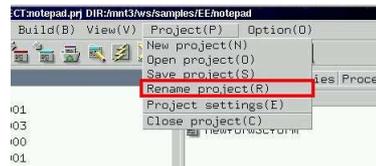
Select [Save project] of the [Project] menu to save.



[Saving a project]

3.4 How to rename project and save it

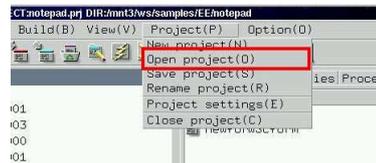
Select [Rename project] of the [Project] to rename project and save it. Then a file selection dialog is indicated, and you can specify the file name which is used as new project name.



[Renaming project]

3.5 How to open a created project

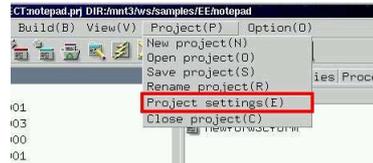
Select [Open project] of the [Project] menu to open a created project.



[Opening a project]

4 How to set environment of project

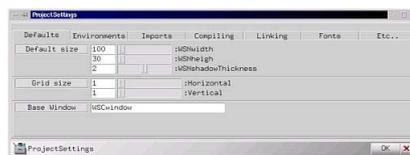
Select [Project settings] of the [Project] menu to set environment of project.



[Menu of project settings dialog]

You can set environments like the followings.

- Default value
Specifies default values of the instance such as dimension and so on.
- Tools
Specifies a terminal, a web browser, a source code editor.
- Import libraries
Specifies import libraries which contain new GUI objects. See section: [How to import libraries which supply new objects]
- Compiler options
Specifies options for compiler, header path, which compiler to use, mode of compiling.
- Linker options
Specifies options for linker, library path, which library to link, which linker to use, which debugger to use.
- Font settings
Specifies fonts.



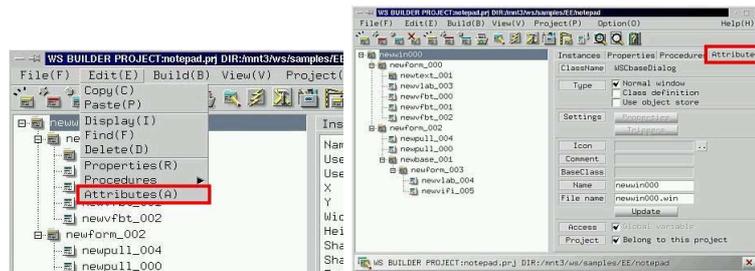
[Project settings dialog]

If you set each item, push a button: [OK] to reflect it. The settings are stored into file: [project-name].prj which is text file, so you can see it.

5 How to add an application window to project

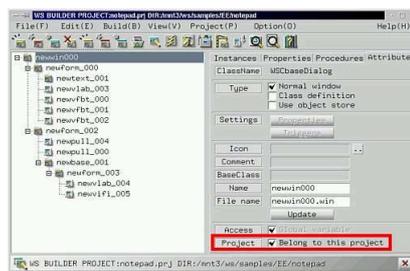
You can add an application windows to project An added application window is loaded together when project is loaded.

Select an application window to add to,and select [Attributes] tab of the inspector.



[Displaying of window attributes]

Check a radio button:[Belong to this project] as the followin figure.



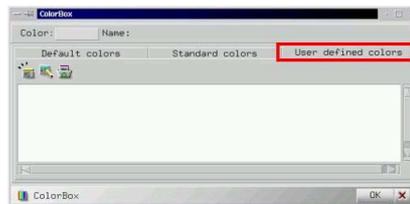
[Add an application window to project]

Registration informatin of appliation window is stored into a file: [project-name].prj which is text file, so you can see it.

6 How to add a color

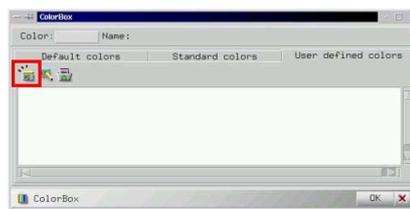
6.1 How to add a color

You can add colors to color selection dialog. Select [Colors] of the [Option] menu.



[Displaying of color selection dialog]

Push the following icon to add an color.



[Displaying of color editor]

You can set color by R,G,B value or sliders or colormap on color editor. if you specify a color,push a button:[OK].



[Color editor]

7 How to edit a color

Select a color which you want to change on color selection dialog, and push the following icon, You can change only user defined color.



[Editing a color]

8 How to delete a color

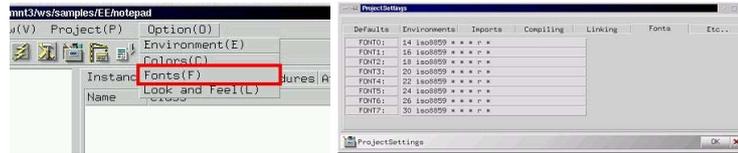
Select a color which you want to delete on color selection dialog, and push the following icon, You can delete only user defined color.



[Deleting a color]

9 How to set up fonts

You can set up fonts of project. Select [Fonts] of the [Option] menu, push a font button which you want to change.



[Displaying the font settings dialog]

Choose a your favorite font. The following figure is the case of UNIX.



[Choosing a font]

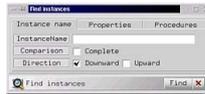
Select the following item.

- 1. font height: dots
- 2. font vender
- 3. font code
- 4. font family
- 5. font weight
- 6. font slant

10 How to find application windows and instances

You can find application windows and instances in project with key such as it's name,value of property, event procedure.

Select [Find] of [Edit] menu.

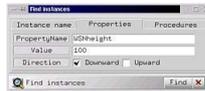


[Search of instances]

There are three ways to search instances as follows.

- Search by instance name
You can appoint instance name as search key.
- Search by property value
You can appoint property value as search key.
Appoint the property with the format of WSNxxxxxx, and a value.
- Search by procedures
You can appoint procedure name ,triggers or function name as search key.

If the application builder find some instances,indicates it highlighted in the inspector,and with [Find] button for a next one.



[by property]



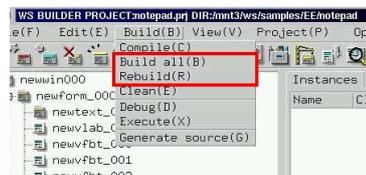
[by procedure]

G Compiling/Building

1 How to build project

1.1 How to build project

You need to build to execute your application. The application builder builds project automatically, and generates an executable. Select [Build all] or [Rebuild] of the [Build] menu, [Rebuild] cleans project: (deletes object files, executable), and compiles it from the first.

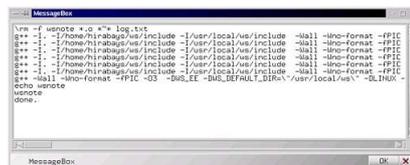


[Building a project]

- Build all
The application builder compiles only modified files, so it is done speedy.
- Rebuild
The application builder compiles all files. if a header file which is included by most files is modified, it requires [Rebuild].

When compilation begins, compilation window is indicated as follows. it shows information from the compiler and linker.

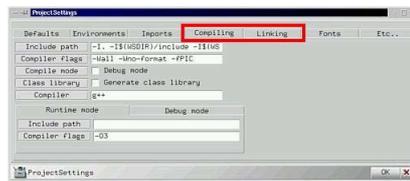
You can confirm whether there are compile errors.



[A view of compilation window]

1.2 How to appoint header path and library path

You can appoint header path, mode of compilation, specification of flag, a library to link and so on. Select [Project settings] of the [Project] menu and Select [Compiling] or [Linking] tab of the following dialog.



[Settings for the compiler and linker]

You can appoint the following items.

- Include path
Appoint path of header files. In case of an ordinary compiler of UNIX, You can appoint it as follows.

-Ipath1 -Ipath2 ...
- Compiler flags
Appoint flags for compiler.
- Compile mode
You can select normal mode or debugging mode.
- Class library
You can select whether it generates normal executable or class library (shared library).
- Compiler
You can appoint your favorite compiler.
- Libraries

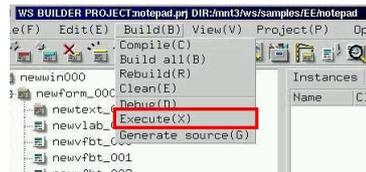
Appoint path of libraries for linking. In case of an ordinary linker of UNIX, You can appoint it as follows.

-Lpath1 -Lpath2 ... -llibrary1 -llibrary2 ...

- Linker flags
Appoint flags for linker.
- Linker
You can appoint your favorite linker.
- Debugger
You can appoint your favorite debugger.

2 How to execute a compiled load module

If compilation is successful, you can execute it. Select [Execute] of the [Build] menu.



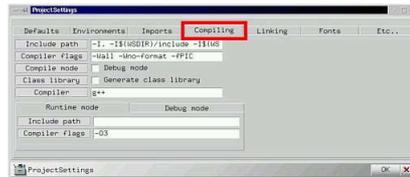
[Executing an application]

Select [Abort] of the [Build] menu which is the same position.

3 How to appoint compiler options

3.1 How to appoint compile options

If you want to appoint for compiler options, select [Project settings] of the [Project] memnu, and push [Compiling] tab of the window.



[Settings about compiler]

You can appoint the folloing items.

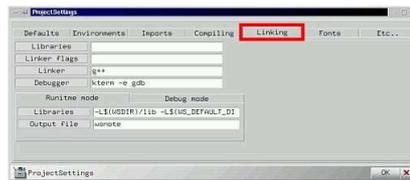
- Include path
Appoint path of header files. In case of an ordinary compiler of UNIX, You can appoint it as follows.

-Ipath1 -Ipath2 ...
- Compiler flags
Appoint flags for compiler.
- Compile mode
You can select normal mode or debugging mode. Debugging mode is for debugging application with debugger.
- Class library
You can select whether it generates normal executable or class library (shared library).
- Compiler
You can appoint your favorite compiler.
- Runtime mode
You can apoint options for normal mode only.
- Debug mode
You can apoint options for debugging mode only.

4 How to appoint libraries to link

4.1 How to appoint libraries to link

Select [Project settings] of the [Project] menu to appoint linking options such as libraries to link.



[Linking options]

You can appoint the following items.

- Libraries

Appoint path of libraries for linking. In case of an ordinary linker of UNIX, You can appoint it as follows.

```
-Lpath1 -Lpath2 ... -llibrary1 -llibrary2 ...
```

- Linker flags
Appoint flags for linker.
- Linker
You can appoint your favorite linker.
- Debugger
You can appoint your favorite debugger.
- Runtime mode
You can appoint options for normal mode only.
- Debug mode
You can appoint options for debugging mode only.
- Output file
You can appoint file name of executable.

5 Source file addition setting

5.1 Source file addition setting

You can add any source files to the project. Selecting [Project setting] in [Project] manu and select compile. Then, list object name in the source code which you want to add in the column of [Additional Object]

For example, when src1.c,src2.c,src3.c are to add to the project, please specify as follows:

```
src1.o src2.o src3.o
```

6 How to appoint debugging mode

You can build with debugging mode. the mode is for effective debugging with debugger.

Select [Project setting] of the [Project],and push [Compiling] tab, make [debug mode] checked.

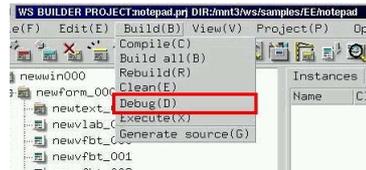
If you changed the mode,please rebuild current application.



[Debugging mode]

7 How to debug an application

You can debug your application with a debugger. Build application with debugging mode, and select [Debug] of the [Build] menu.



[Starting the debugger]

I try to easily describe usage of gdb here. At first, input "run" to execute an application. input "where" to refer to stack trace and see where is, when the program has stopped by error. You can see position of the source by input of "list" that error was generated.

There are the following commands in gdb.

- run
Starts the application.
- cont
Continues the application which is interrupted.
- step/next
Run the application step by step which is interrupted.
- where
Shows stack trace of function call.
- print
Shows value of variables or functions.
- list
Shows the current position of source.
- break
Adds an break point. You can input it as follows.
break file.cpp:XXX
break CLASS::FUNCTION()
XXX is a line number.
- cntl-C
Interrupts the application.

```

gdb
#2 0x4030e4e in __VAllocNameOfColor ()
#3 0x401049e in WSColor::initialize (this=0x0061453) at WSColor.cpp:116
#4 0x4010524 in WSColor::setColorName (this=0x0061458,
  cname=0x4012576c "Hffffff") at WSColor.cpp:72
#5 0x40005713 in WSColorSet::setStaticColorName (this=0x004cc18,
  colname=0x4012576c "Hffffff") at WSColorSet.cpp:144
#6 0x40005dc1 in WSColor (colname=0x4012576c "Hffffff")
  at WSColorSet.cpp:272
#7 0x4007ba26 in WSChase::setPropertyJ (this=0x00613d0,
  prop_name=0x40125751 "ForeColor", value=0x4012576c "Hffffff")
  at WSChase.cpp:1314
#8 0x401042c4 in WSChaseDialog::initialize (this=0x005a2f0)
  at WSChaseDialog.cpp:293
#9 0x40107307 in WSFontSelect::initialize (this=0x005a2f0)
  at WSFontSelect.cpp:138
#10 0x40106708 in _sfonts_init_0 at WSFontSelect.cpp:46
#11 0x40008d4c in WSAppDev::execInitHandler (this=0x004fa80)
  at WSAppDev.cpp:101
#12 0x401133a7 in main (argc=1, argv=0xfffff040) at Main.cpp:43
#13 0x40220c13 in __libc_start_main (main=0x40220c13, argc=1,
  argv=0xfffff040, init=0x00492554, fini=0x004a4cdc, _fini=,
  ---type (return) to continue, or q (return) to quit---
  rtld_fini=0x40000250, stack_end=0xfffff00c)
  at ../sysdeps/generic/libc-start.c:70
(qdb) break WSChase.cpp:400
Breakpoint 2 at 0x4007836: file WSChase.cpp, line 400.
(qdb) step
Single stepping until exit from function XcmsLRGB_ParseStrings,
which has no line number information.
0x4032c70 in _XcmsParseColorString ()
(qdb) next
Single stepping until exit from function _XcmsParseColorString,
which has no line number information.
0x403236e in _XcmsResolveColorString ()
(qdb)

```

[Debugging the application]

8 How to use a trace debugger

You can use trace debugger the execution status of event procedures that you created. First of all, build the application, and select [Trace Execution] in [Build] menu. Then, trace dialog appears as follows:



[Trace Dialog]

And push [start] button on the trace dialog to start tracing. When event procedure is activated, the followings are output in the trace dialog. When you want to stop tracing, please push [stop] button.



[Trace output]

If the function existing an event procedure,

```
function function-name( ) end
```

is not output in the trace log and the application terminates abnormally, you can expect fault is in the event procedure.

Further more, you can see a trace by application using WSGFtrace() function as follows(A):

```
\#include <WScom.h>
\#include <WSCfunctionList.h>
\#include <WSCbase.h>
//-----
//Function for the event procedure
//-----
void btn\_ep1(WSCbase* object){
```

```
object->setProperty(WSNlabelString, "Hello!");  
WSCstring string;  
string = "btn\_ep1 called!!!!\n";  
WSGFtrace(string);                               //(A)  
}  
static WSCfunctionRegister op("btn\_ep1", (void*)btn\_ep1);
```

Note that output by WSGFtrace() is enabled only when the tracing is enabled.

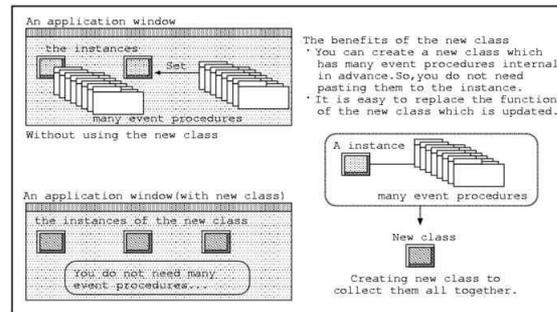


[Trace output by an application]

H Class Application Window

1 What is class application window?

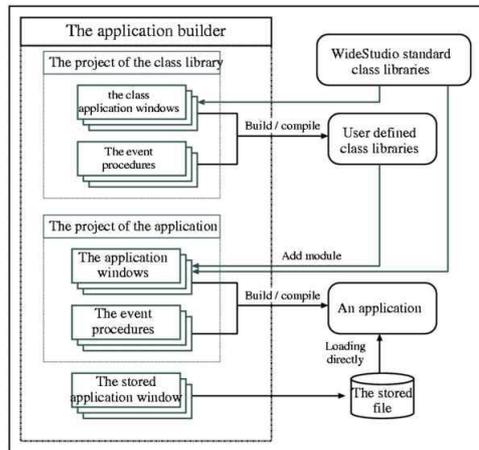
You can make more complicated object afresh on the bases of C++ object class for WideStudio.



You can get the template class which is generated by the application builder automatically, and which inherits the object class of WideStudio, or consists of them. Developers define a new class by adding new member function to it.

It is class application window that supports easy development of new object class.

- Development which is same as normal application window
You can create and edit an class application window with same method of normal one.
- Defintions of new property
You can add a new property to the new class by dialog. and the application builder generates source code about that automatically.
- Class library construction
The application builder can builds an shared library of your new classes which can be used by the other project.



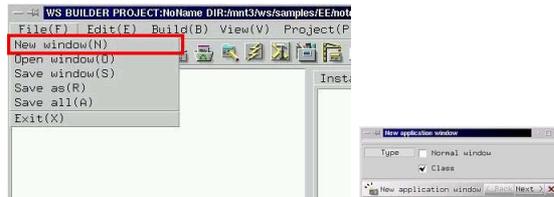
The followings shows generate style of the class.

Application window descendant	Makes an application window a class.
Application window part descendant	Makes a part of application window a class.
One class descendant	makes a new class from a class.

2 How to create a new class application window

2.1 How to create a new class application window

Select [New window] of [File] menu to pop up the application window wizard dialog, and check the [Class] field.



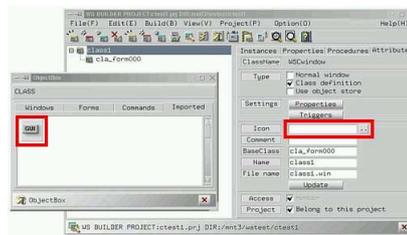
[A new class application window]

2.2 How to appoint an icon and title string of the class

The class application window can be packed as class library and can be used by the other project. When, project load the library, the classes which are in it appears in the [Imported] section of the object box dialog.

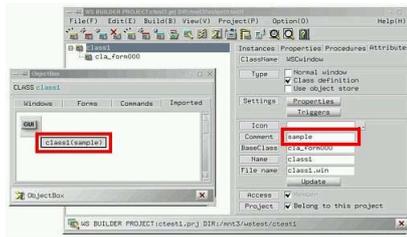
You can appoint an icon which is indicated in the object box and an title string which is used by balloon help of it. If you do not appoint it, the default icon and string are used.

It is required that icon is xpm or jpg format. The following figure shows the default icon(left of figure), and an input field to appoint an icon file(right of figure).



[The default icon and appointment of icon]

The following figure shows the default title string(left of figure) of a balloon help, and an input field to appoint title string file(right of figure).



[Appointment of title string of the balloon help]

3 How to appoint class base for a new class

3.1 Default base class

By default, the class of top window of class application window becomes a base class for a new class. The application builder generates the following c++ source codes. [classwin] is name of the class application window.

classwin .h

A public header file for the class. You can add new members to this.

classwin .cpp

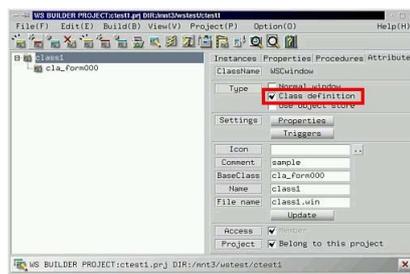
A public source file for the class. You can add new codes to this.

classwin P.h

A private header file for the class. You do not edit it.

classwin P.cpp

A private source file for the class. You do not edit it.



[Definition of class application window]

A sample "class1" of class application window in the figure derive from WSCwork window which is a top window of it and becomes a c++ class "class1". Top window is used as base class by default, and you can appoint other base class with [Attributes] section on the inspector.

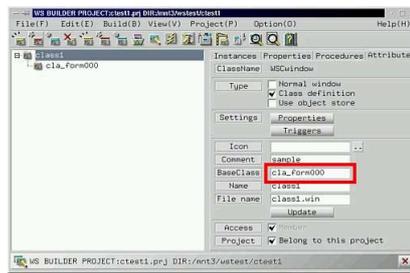
3.2 How to appoint a base class for a new class

For example, a part of class application window, "cls_form000" a form, I appoint it as a base class here. By default, it uses top window as base class, but if some instance on the class application window is appointed, it becomes a base class.

(Notice) It is required that re-generating source code, if you change the base class. Delete the following file even if edited. [classwin] is name of class application window.

classwin .h

classwin .cpp



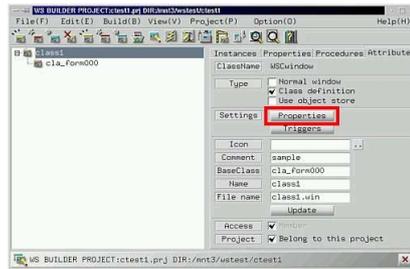
[Appointment of base class]

In this case, a new class:"class1" is created from the instance:"cla_form000" which is appointed as base class. So, the class:"class1" inherits WSCform. If there are children on the instance for the base class, they are included together in the new class.

4 How to add a new property / edit / delete it.

4.1 How to display property setup dialog.

You can add a new property to your class application window with property setup dialog. Pop up the dialog by clicking the following button.



[Property setup dialog]

4.2 How to add a new property

Click the following icon to display property create dialog.



[Creating a new property]

There are the followings elements to appoint.

- Property name
Appoint a property name begins "WSN...".
- Property type
Appoint a data type of variable of property.
- Variable name
Appoint a variable name for property. The property uses it to store data.
- Default value
Appoint a default value of property.
- Attribute
There are following attributes.

Kind	Description
Normal	Normal property.
Invisible	Property which is invisible from builder.
Delete	Deletes/Disables an existing property.
Default Change	Changes default value of an existing property.
Ethelize	Make an existing property invisible from builder.

- Builder title
Appoint a property title which is indicated on inspector such as the following.



[A view of property create dialog]

4.3 How to edit a property

Select a property and click the following icon to edit it.



[Editing of a property]

4.4 How to delete a property

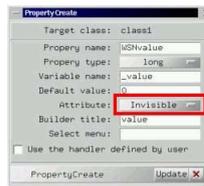
Select a property and click the following icon to delete it.



[Deleting of a property]

4.5 How to create a new invisible property

You can create a new property which is invisible from application builder. Select [Invisible] attribute with property create/edit dialog or property.



[Create a invisible property]

5 How to delete/make invisible an existing property which is inherited

5.1 How to delete an existing property

You can delete an existing property which is inherited from base classes. A new property which has same name and has delete attribute deletes it. Display property create dialog.



[Displaying the property create dialog]

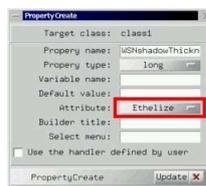
input a name which is same as an existing property, select delete attribute. Property type, Variable name, Default value, and Builder title are not required. For example, it deletes "WSNshadowThickness" property.



[Deleting a existing property]

5.2 How to make an existing property invisible

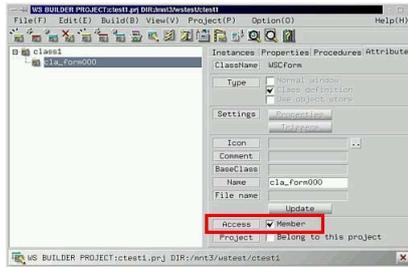
You can make an existing property invisible which is inherited from base classes. A new property which has same name and has ethelize attribute deletes it. Display property create dialog and input a name which is same as an existing property, select ethelize attribute. Property type, Variable name, Default value, and Builder title are not required. For example, it make "WSNshadowThickness" property invisible.



[Making a existing property invisible]

6 How to make child instances be member of the class

You can define the child instances as member of the class. Select a child on the inspector, and push [Attributes] tab, check radio button: [Access].

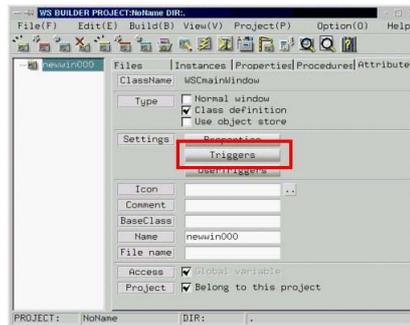


[Definition of member]

7 How to add/edit/delete triggers

7.1 How to display the trigger setup dialog

It is possible to add triggers to your class application window with trigger setup dialog. Click the following icon to display the trigger setup dialog.



[The trigger setup dialog]

7.2 How to add triggers

Select a trigger to add with following menu, click [Add] button, and update data by [OK] button.



[Adding triggers]

7.3 How to delete added triggers

Select an added trigger on the list, and click [Delete] button, and update data by [OK] button.

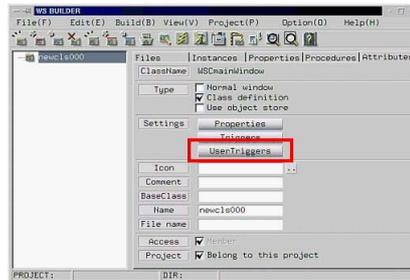


[Deleting added triggers]

8 How to add/edit/delete a new user trigger.

8.1 How to display the user trigger setup dialog

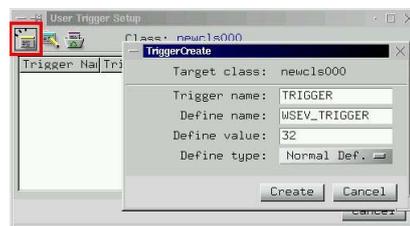
It is possible to add user defined triggers to your class application window with the user trigger setup dialog. Click the following icon to display the user trigger setup dialog.



[The user trigger setup dialog]

8.2 How to add user triggers.

Select a following icon and, input new user trigger name, its value, and click the create button.



[Adding user triggers]

8.3 How to delete added user triggers.

Select an added user trigger in the list, and click the following icon to delete it.

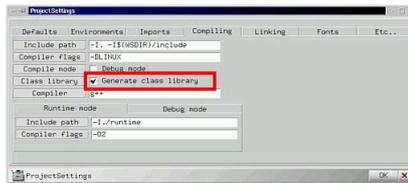


[Deleting added user triggers]

9 How to create a class library

The application builder has a function to create the class library which includes your classes. The output of project is usually a executable, but you can appoint a mode to turn it into a class library which is a shared library.

Check the radio button:[Class library] to generate output of project as a class library. The following figure is section:[Compiling] of project settings dialog.



[Generating a class library]

The following [output] is the output file name which is appointed with section:[Linking]

System	library name
UNIX	lib[output].so (shared library)
	lib[output].a (static library)
Windows-NT	lib[output].dll (dynamic link library)

See section:[How to use the new classes of the import libraries] to use these libraries from other project.

I Stored Application Window

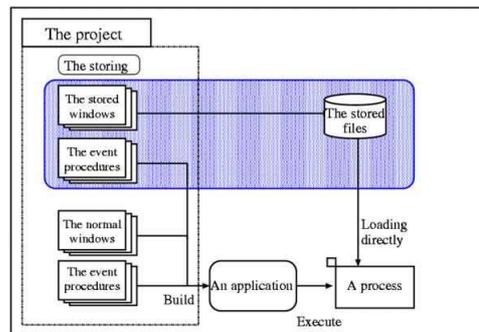
1 What is Stored Application Window?

You can store your application window with the application builder. it is a "stored application window". An application of WideStudio has a function to load an application window directly from file during execution.

This function realizes the followings.

- Saving of memory,Speeding up starting
Application do not contain all of the application windows at starting to save of memory and speed up. and display them with loading from stored files.
- Switching a part of application window
Application can load the partial stored windows as data, and can display it on its window as a part of the window. This function realizes to display and switch figures like an "Map", "Drawing sheet" which is partial stored window on a window of application.
- For easy maintenance
You can share the stored window with some applications, and you can edit it directly, and the window of the application can be updated with no compiling. It is a merit for maintenamce of application.

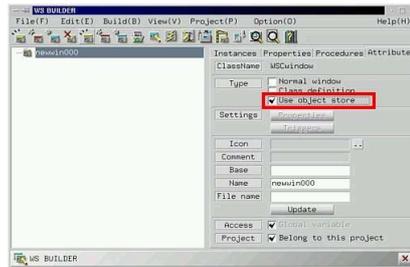
You can store an application window with application builder. In addition to it,you can make the application window which is belong to project be stored attribute, then the application builder builds an executable which excludes it and stores it automatically. The executable load them with WSGFloadWindow() function individually during execution.



2 How to make an stored application window

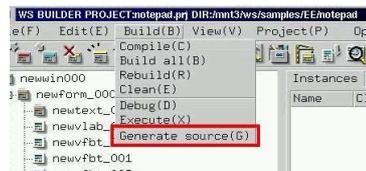
2.1 How to save an application window by stored type

You can save a normal application window by stored type. Select it with the inspector, check the radio button:[Use object store] of section:[Attribute].



[Be stored]

Next, select [Generate source] of the [Build] menu as follows.



[Generate a stored application window]

2.2 How to add the stored application window to project

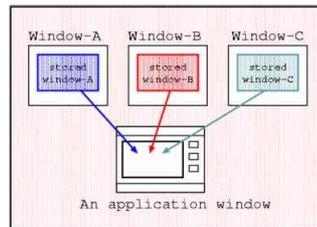
We do not hope to store it manually when build the executable. You can select stored attribute for each application window of project with application builder, Select it with the inspector, check the radio button:[Use object store] of section:[Attribute]. It is same as above method to make it be stored attribute. The application builder does not compile it, builds an executable which excludes it and stores it to object store file.

The file is [window name].oof.

The executable load them with WSGFloadWindow() function individually during execution. See section:[How to load the stored application window directly from the program] of Programmer's Guide.

3 How to make a partial stored application window

You can make a part of application window be stored. Application can load the partial stored window and display it on its window. A merit of partial stored windows is that application can display them next to next by switching on a same window.



[Displaying partial stored windows]

For example, there are partial stored application window A, B, C which store base is shaded coloring area in the figure. You can load them and display A or B or C on the application window.

Select [Attributes] tab of the inspector and appoint store base instance to the following field to store partial.

if this field is blank, this case is Default, top window is used as store base instance.



[Appointment of store base instance]

See section: [How to load a stored application window directly] of Programmer's Guide.

J Remote Instance

1 What is the Remote Instance

WideStudio enables an instance existing in the WideStudio application running on other machine to be called in the same way as if it existed in own process. The remote instance feature covers the following:

- Distributed Computing

An instance(object) existing in another WideStudio application can be accessed in the same way as an instance(object) which exists in own process to enable distributed computing easily. Large scale WideStudio applications can be designed with being functionally separated in multiple processes to operate collaborating with each other.

- Seamless distribution over the network

Instances scattered over WideStudio applications running on multiple computers on the network can be seamlessly referred to without aware of their whereabouts. Agents running on each computer manage remote instances existing on their platform and exchange information among the agents, by which any remote instance can be managed its whereabouts.

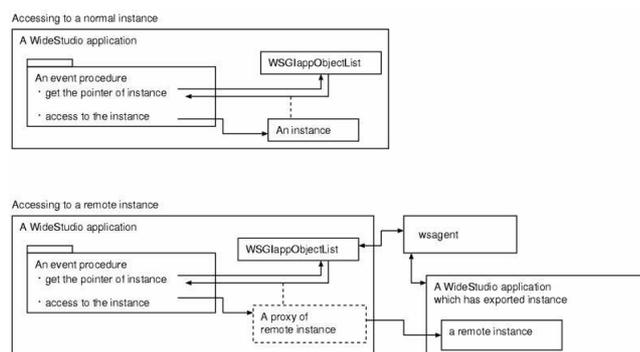
WideStudio applications can access such remote instances without being aware of their whereabouts to plug into correct remote instances automatically.

- Fault tolerance improvement

WideStudio applications can be run multiplexing. For example, when one application out of the multiplexed goes down, an agent detects it and interchanges the remote instance on another application.

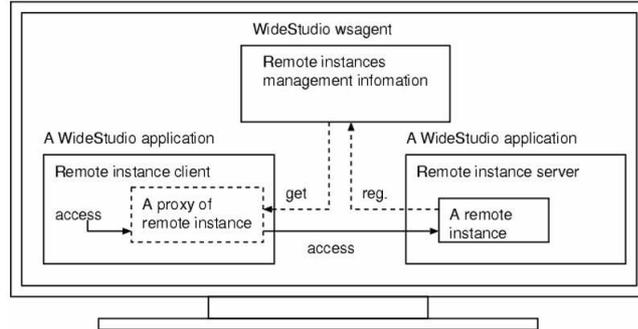
As for the WideStudio application, it can access the remote instance without aware of its being exchanged by multiplexing

Accessing to a remote instance is performed by using the remote instance acquired with an object management in the same way of accessing to an usual instance acquired by WSGIappObjectList() object management.



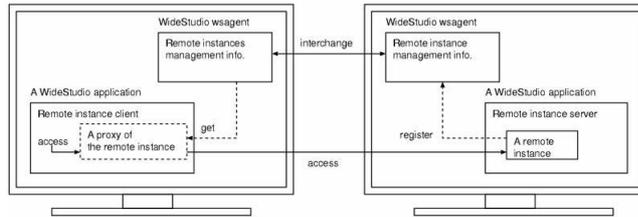
In calling a remote instance existing on the same machine, the agent running on the same machine gets a remote instance to be called.

Accessing to the instance on local machine



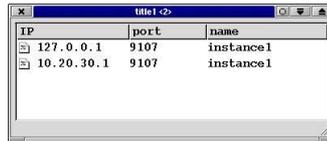
In calling a remote instance existing on a different machine, agent running on each machine exchanges information of remote instances to get it be called.

Accessing to a remote instance



2 How to start up Agent

An agent plays an important role in realizing a remote instance feature. It provides whereabouts information on remote instances against the client who wants to access that instances. An agent 'wsagent' should be run on a machine when a remote instance feature is used.



IP	port	name
127.0.0.1	9107	instance1
10.20.30.1	9107	instance1

[Agent]

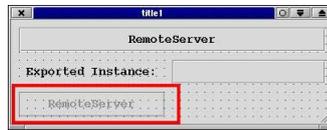
3 How to construct a remote instance server

A remote instance server has no difference from an usual WideStudio application and is to publisize as remote usual instances exiting within applications

By publisizing as remote, the publisized instances are registered on an agent automatically to be accessed from external applications

3.1 How to get an appliation become a remote instance server

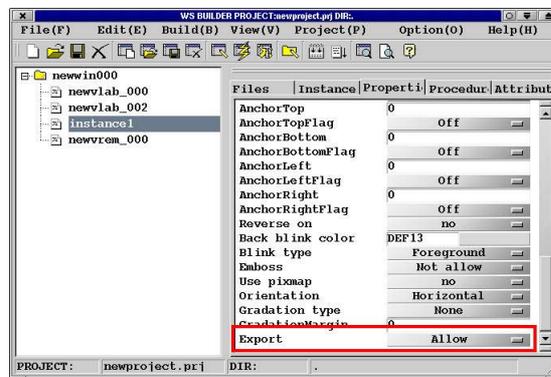
In order to get a WideStudio application a remote instance server, WSCvremoteServer class is used. An application can become a remote instance server by placing a WSCvremoteServer class instance in Object Box's NonGUI section on any application window.



[WSCvremoteServer Placing]

3.2 How to publisize an instance as a remote instance

When you want to publisize an instance existing a WideStudio application to external, turn "Export" of that instance property enable.



[Publisizing an instance as a remote instance]

3.3 How to start up an agent

Starting agent who manages remote instances that are publisized and existing in a remote instance server, remote instances can be accessed. Agent can be started by wsagent command.



IP	port	name
127.0.0.1	9107	instance1

[Starting up agentd, "wsagent"]

4 What is wsagent

WideStudio application can access remote instances existing in a remote instance server in the same way as instances existing on own process.

Fist of all, in order to access remote instances, you will need to know whereabouts of the remote instance on a machine by asking it to an agent.

An agent manages remote instances existing on a remote instance server.

5 Before accessing remote instances

Before accessing a remote instances, WSCvremoteClient class is used. WSCvremoteClient class supports communication to an agent who manages remote instances

Only action to communicate with an agent is to place WSCvremoteClient existing in Object Box's NonGUI section on given application widow of the applicatoin.

6 How to access remote instances

Specifying a name, a remote instance can be accessed via a virtual remote instance got by object management. Please refer to Remote Instance Edition in the Programming Guide for more details