HiGVBuilder

User Guide

Issue **04**

Date **2015-07-25**

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

About This Document

Purpose

The HiGVBuilder is a graphic editing tool developed based on the Eclipse rich client platform (RCP). The HiGVBuilder works with the HiGV component to enable you to edit graphical user interfaces (GUIs) and convert edited GUIs into XML files by performing visible operations. The core framework technologies include the graphical modeling framework (GMF), graphical editing framework (GEF), Eclipse modeling framework (EMF), and C developer tools (CDT).

Related Version

The following table lists the product version related to this document.

Software	Version
HiGVBuilder/Xml2Bin	V200R002C00SPC02 or later

Intended Audience

This document is intended for:

- UI design engineers
- Software development engineers

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 04 (2015-07-25)

This issue is the fourth official release, which incorporates the following changes:

Section 3.3.3.2 to section 3.3.3.27 are deleted.

Issue 03 (2015-05-25)

This issue is the third official release, which incorporates the following changes:

The entire document is updated.

Issue 02 (2014-09-05)

This issue is the second official release, which incorporates the following changes:

Chapter 3 Editing the GUI

Sections 3.3.3.25 and 3.3.3.26 are added.

Issue 01 (2013-11-30)

This issue is the first official release.

Contents

Figures

Tables

1 Overview

1.1 Introduction to the HiGVBuilder

The HiGVBuilder is a graphic editing tool that allows you to edit GUIs and generate the codes and the .xml binary files by using the xml2bin tool. The generated codes and files can be identified by the HiGV component by performing visible operations. HiGV is a lightweight single-process GUI system that provides unified lightweight, efficient, and easy-to-use GUI solutions for HiSilicon chip platforms.

NOTE

Before using the HiGVBuidler, you are advised to know about the HiGV by referring to the *HiGV Development Guide*.

The HiGVBuilder provides the following functions:

- Edits GUIs.
- Manages projects and project resources.
- Generates binary interface files for the HiGV component.

1.2 Environment Preparations

1.2.1 Hardware Environment

The following describes the minimum hardware configurations for running the HiGVBuilder:

- CPU: Pentium dual-core 2.0 Hz (Pentium dual-core 3.0 Hz or higher recommended)
- Memory capacity: 1 GB (2 GB or larger recommended)
- Available hard disk space: 500 MB (1 GB or larger recommended)

1.2.2 Software Environment

The following is the dependent software environment of the HiGVBuilder:

- OS: Windows XP/Windows 7
- Software: Java Runtime Environment (JRE) 1.6 or later

1.3 Getting Started

The HiGVBuilder is green software that does not need to be installed. To run the HiGVBuilder, decompress **HiGVBuilder zip**, and double-click **HiGVBuilder.exe**. Because the HiGVBuilder is developed based on the Eclipse RCP, you must install JRE 1.6 or later before using the HiGVBuilder.

1.3.1 Installing JRE

If JRE 1.6 or later is not installed on your PC, perform the following steps to install it before using the HiGVBuilder:

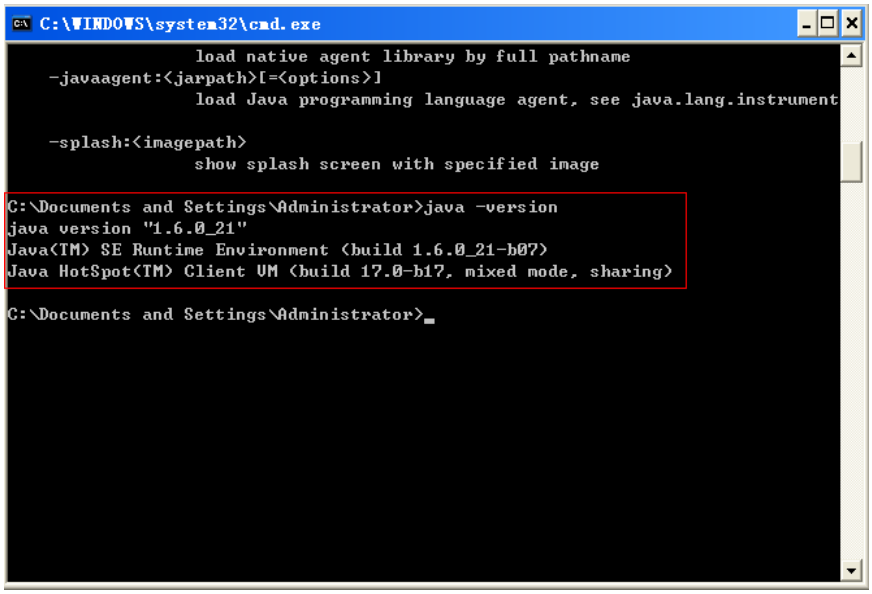
Step 1 Preinstall JRE 1.6 (jre-6u1-windows-i586-p); otherwise, the HiTool may fail to run properly.

You can download JRE 1.6 from http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase6-419409.html.

Step 2 Double-click the decompressed or downloaded JRE installation package, and install JRE by following the wizard.

After the installation completes, you need to check whether JRE is successfully installed. Choose **Start** > **Run**, enter **cmd**, click **OK**, and enter **java -version** in the displayed command-line interface (CLI). If the information shown in Figure 1-1 is displayed, JRE is successfully installed.

Figure 1-1 Checking whether JRE is successfully installed



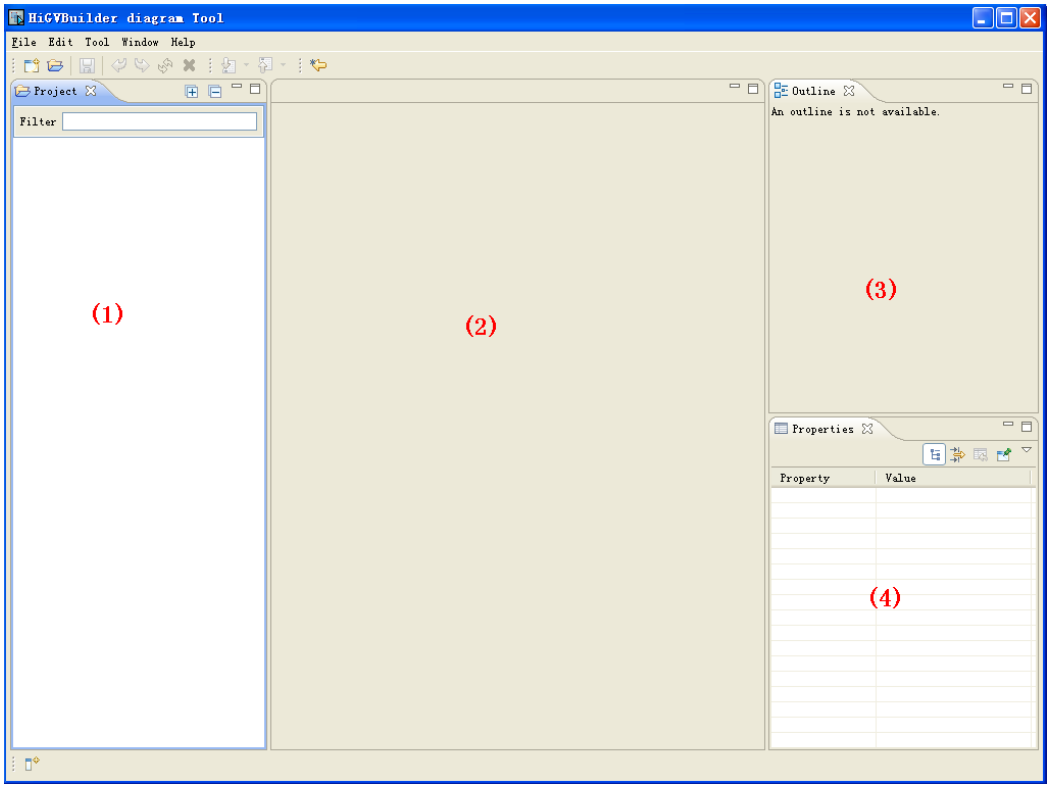
----End

1.3.2 Main GUI

This section describes only the basic functions of the HiGVBuilder to help HiGVBuilder users rapidly get familiar with it. For details about the functions, see the following chapters.

Figure 1-2 shows the main GUI of the HiGVBuilder.

Figure 1-2 Main GUI of the HiGVBuilder



- Area 1 (Project Explorer): Allows you to manage all project resources such as drawings, skins, fonts, data mode, and multi-language character strings.
- Area 2 (editing area): Allows you to edit drawings and common resources.
- Area 3 (Outline): Allows you to display the thumbnails and control hierarchy.
- Area 4 (Properties): Allows you to modify control properties when editing drawings.

1.3.3 Opening a Project

To open a created project file, you can click the **Open Project** button on the toolbar shown in Figure 1-3, or right-click **Open Project** (for details about how to create a project, see section 1.3.4 "Creating a Project"). Take the STB Linux platform Halo_V1.0.2.0 as an example. The preset HiGVBuilder project file **X.hgp** is in the **project\Hi37XX_X_X\higv** directory. You can edit it by directly opening the STB platform GUI.

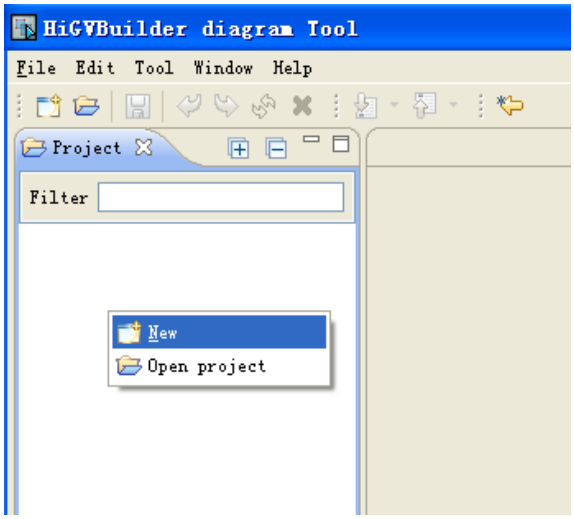
Figure 1-3 Open project button

1.3.4 Creating a Project

To create a project, perform the following steps:

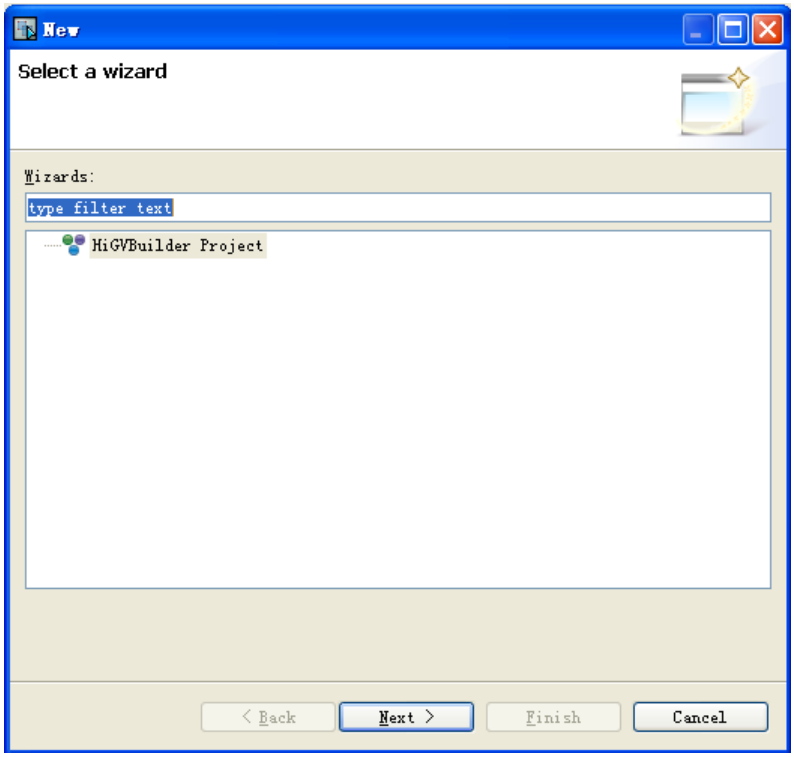
Step 1 Right-click the blank area in the **Project Explorer** pane, and choose **New** from the shortcut menu, as shown in Figure 1-4.

Figure 1-4 Creating a project



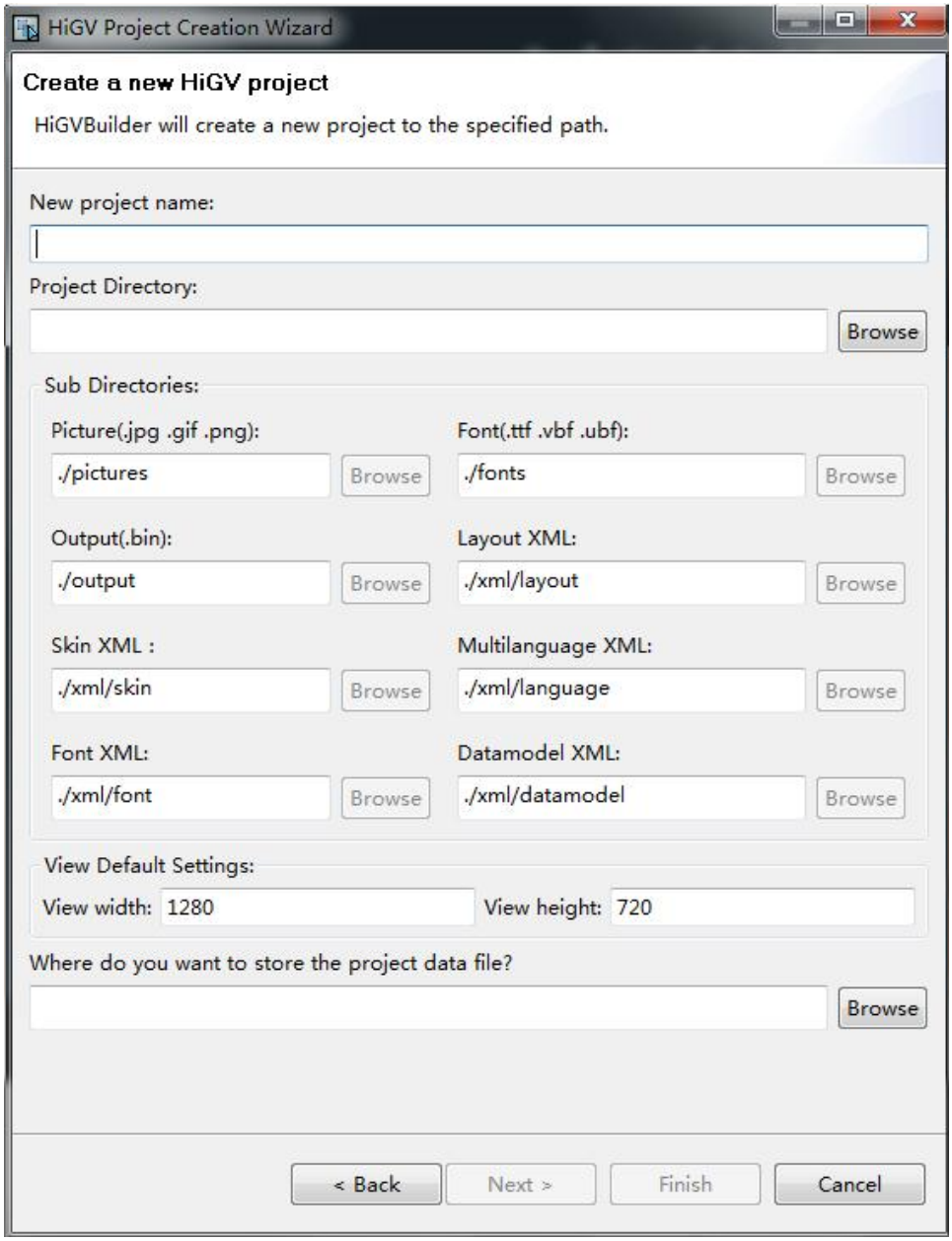
The **New** dialog box shown in Figure 1-5 is displayed.

Figure 1-5 New t dialog box



Step 2 Select **HiGVBuilder Project**, and click **Next**. The **HiGV Project Creation Wizard** dialog box is displayed as shown in Figure 1-6.

Figure 1-6 Creating a project



Step 3 Enter the project name in the **New project name** text box. Click **Browse** next to **Project Directory** to select the directory of the current project, which needs to contain all project resources. Select the directory for storing the project description file (this directory can be different from the current project resource directory). Use the default values for other parameters. Click **Finish** as shown in Figure 1-6.

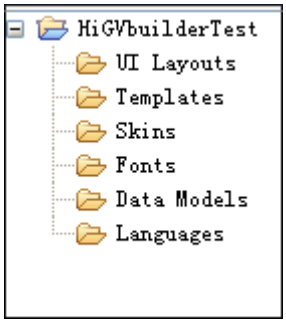
Note the following:

- **Picture**: sub-directory for the picture resources
- **Font**: sub-directory for the font resource files
- **Output**: sub-directory for the output files
- **Layout XML**: sub-directory for the GUI .xml description files

- **Skin XML**: sub-directory for the skin .xml description files
- **Multilanguage XML**: sub-directory for the multi-language .xml description files
- **Font XML**: sub-directory for the font .xml description files and data model .xml description files
- **View width**: width of resolution
- **View height**: height of resolution

The new project is displayed in the project management area, as shown in Figure 1-7.

Figure 1-7 Directory structure of the new project



----End

NOTE

In the preceding instance, the project name is HiGVBuilderTest.

1.3.5 Create a GUI

The new project has only a structure without contents. You need to create a GUI. For the concepts of related resources, see section 3.2.1 "Introduction."

To create a GUI, perform the following steps:

Step 1 Create a skin XML file. Right-click **Skins**, choose **Create Skins** from the shortcut menu, enter a resource ID as a unique identifier of the resource, and click **OK**. The skin editing GUI is displayed. You can manage skins by using the skin XML file. You can also create multiple skin XML files. For details, see section 3.2.3.2 "Creating a Skin." For more details about the skin resources, see the *HiGV Development Guide*.

Step 2 Create a font XML file. Right-click **Fonts**, choose **Create Fonts** from the shortcut menu, enter a resource ID as a unique identifier of the resource, and click **OK**. The font editing GUI is displayed. For details, see section 3.2.4.2 "Creating a Font."

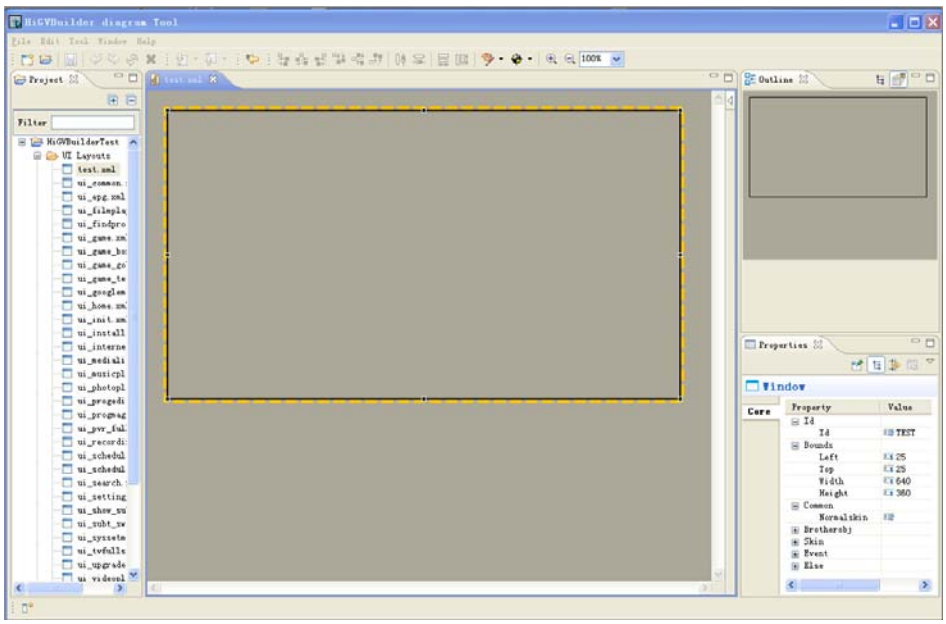
Step 3 Create a data model XML file. Right-click **Data Models**, choose **Create Data Models** from the shortcut menu, enter a resource ID as a unique identifier of the resource, and click **OK**. The data model editing GUI is displayed. For details, see section 3.2.6.2 "Creating a Data Model."

Step 4 Create a multi-language XML file. Right-click **Languages**, choose **Create Language File** from the shortcut menu, and click **OK**. The multi-language editing GUI is displayed. For details, see section 3.2.5.2 "Managing Multi-Language Character Strings."

Step 5 Right-click **UI Layouts**, choose **Create UI Layouts** from the shortcut menu, create a UI XML file in the displayed HiGVBuilder project directory, and click **OK**.

Then the GUI editor is automatically opened, as shown in Figure 1-8.

Figure 1-8 Layout of the new editing GUI



----End

Then you can edit the new GUI.

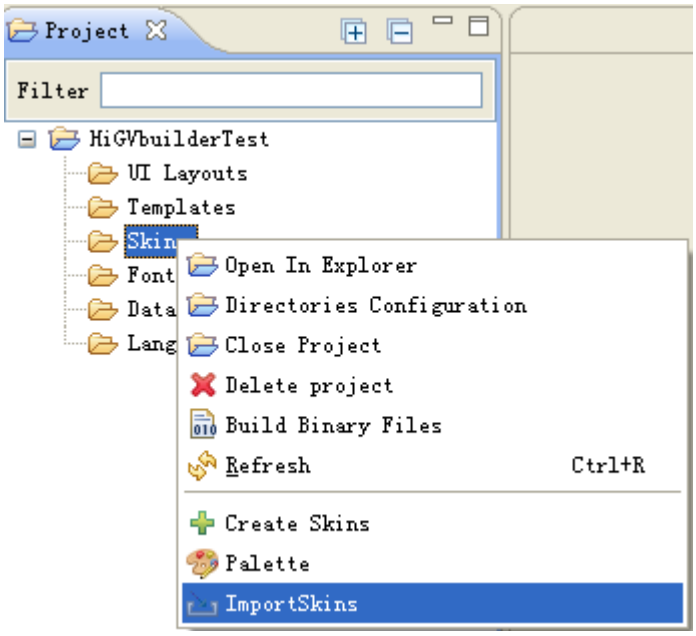
1.3.6 Importing a GUI

If the resource and UI XML files are created, you can import the XML files by performing the following steps:

Step 1 Import the XML files for skins, fonts, data models, and languages. Take the skins as an example. Right-click **Skins**, choose **ImportSkins**, and select the corresponding XML file. The XML file is typically stored in the corresponding skin directory when the project is created, and it must be in the project directory. See Figure 1-9.

Step 2 Import the UI layouts XML file. Right-click the **UI Layouts** directory, choose **Import UI Layouts**, and select the corresponding XML file.

Figure 1-9 Importing skin XML files



NOTE

- All resources in the HiGV are managed by the XML files. For example, you need to set IDs for images and then write the images into the skin XML file.
- All resources and UI XML files are preset in the STB Linux product platform by default. You only need to add resources as required.

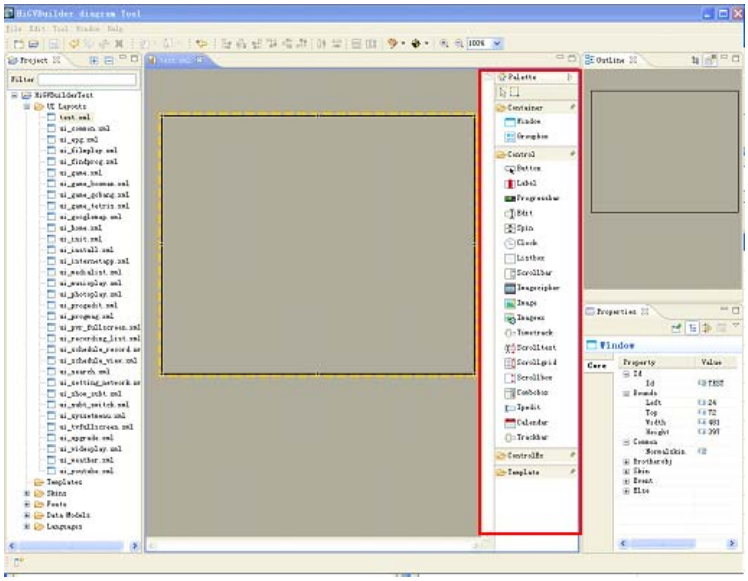
----End

1.3.7 Editing a GUI

The HiGVBuilder allows you to generate controls by dragging or clicking the edit area. To be specific, to generate a control in the edit area, select a control in the red rectangle shown in Figure 1-10, and drag it to the edit area on the left or directly click the edit area.

When a drawing is created, the HiGVBuilder adds a window container for it by default, as shown in Figure 1-10.

Figure 1-10 Window container

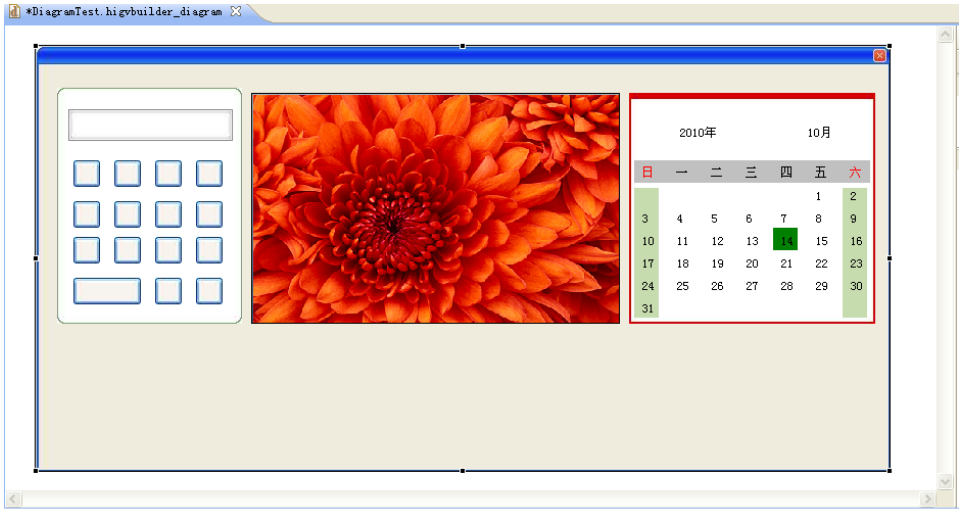


NOTE

The yellow rectangle in Figure 1-10 is a window container.

You can place any controls in the window container. Figure 1-11 shows the GUI with added controls.

Figure 1-11 GUI with added controls

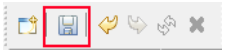


As shown in Figure 1-11, the left part is an interface similar to a calculator, the middle part is a picture frame in which a flower picture is inserted, and the right part is a calendar that displays the current date.

To modify the properties of a control, select the control in the window container, and modify the properties displayed in the Properties pane. For details, see section 2.5.2 "Properties View."

You can also modify controls by dragging them. For details, see section 2.5 "GUI Editor View."

1.3.8 Saving an Edited GUI as an XML File

After editing a GUI, you can click  on the toolbar to save the GUI as an XML file.

NOTE

You can also press **Ctrl+S** to save the GUI as an XML file.

This section briefly describes only some of the HiGVBuilder functions. For details, see chapter 2 "GUI and Functions."

2 GUI and Functions

2.1 Menu Bar

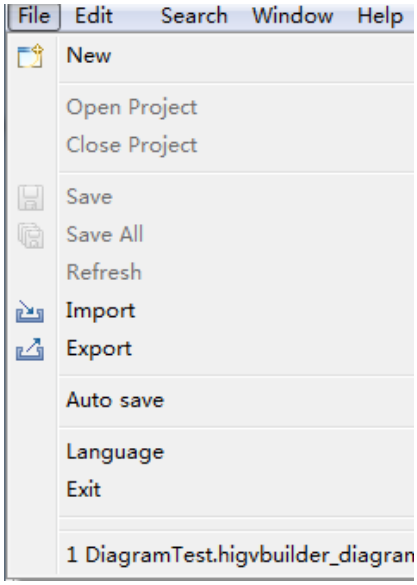
The menu bar contains five main menu items: **File**, **Edit**, **Setting**, **Window**, and **Help**.

Figure 2-1 Menu bar

2.1.1 File

The **File** menu is used to create, save, open, close, import, and export projects and files.

Figure 2-2 File menu



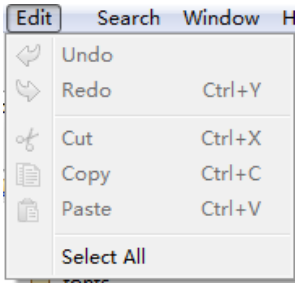
- **New** (**Crtl+N**): Creates a project or file.

- **Open Project**: Opens a project. This option is available when a closed project in the project management area is selected.
- **Close Project**: Closes a project. A project can be closed if it is not used temporarily to release memory and other resources. This option is available when a project in the project management area is selected.
- **Save** (**Ctrl+S**): Saves the current GUI content. It can be used when the content of the current GUI changes.
- **Save All** (**Ctrl+Shift+S**): Saves all the opened GUI contents. This option can be used when the content of the opened GUI changes.
- **Refresh**: Refreshes the project. All files of the project are reloaded from the local working directory. This option can be used to quickly refresh the project after some configurations or files of the project are modified. It is available when a project is selected.
- **Language**: Sets the GUI language of the tool. Currently only Chinese and English are supported.
- **Exit**: Terminates all work and exit the tool. If there are files that are not saved, the tool displays a warning message.
- List of recent files: Files that are opened in the editor recently are listed here to facilitate editing again.

2.1.2 Edit

The **Edit** menu is used to perform common editing operations such as cancel, copy, paste, and delete.

Figure 2-3 Edit menu



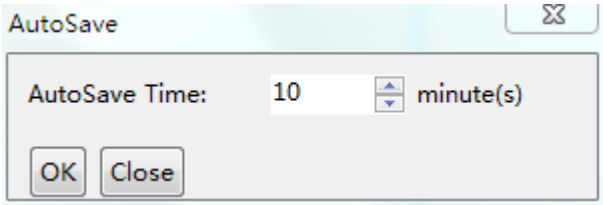
- **Undo** (**Ctrl+Z**): Cancels the recent operation. This option is dedicated for the graphics editor and is available after a GUI is edited. A maximum of 20 consecutive operations can be canceled.
- **Redo** (**Ctrl+Y**): Cancels the recent undo operation. This option is dedicated for the graphics editor and is available after an undo operation is performed. A maximum of 20 consecutive undo operations can be canceled.
- **Cut** (**Ctrl+X**): Cuts the selected content in a drawing or text file to the clipboard.
- **Copy** (**Ctrl+C**): Copies the selected content in a drawing or text file to the clipboard.
- **Paste** (**Ctrl+V**): Pastes the content in the clipboard to the current drawing or text file.
- **Select All** (**Ctrl+A**): Selects all controls in the current editor.

2.1.3 Setting

The **Setting** menu allows you to set some basic parameters of the HiGVBuilder.

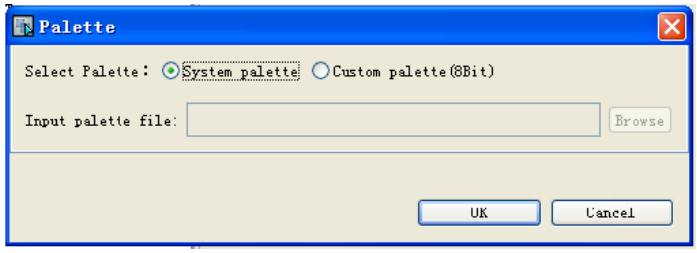
- **Auto save**: Choosing this option displays the dialog box for setting **AutoSave Time**, as shown in Figure 2-4. The default value is 10 minutes.

Figure 2-4 AutoSave dialog box



- **Palette**: Choosing this option displays the dialog box for setting the palette, as shown in Figure 2-5. The default system palette is the Windows 32-bit palette. You can also specify an RLE cropping 8-bit monochrome palette file, for example, **pub.pal**.

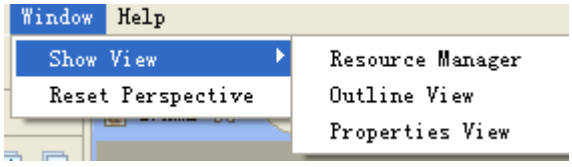
Figure 2-5 Setting the palette



2.1.4 Window

The **Window** menu is used to control perspective drawings and windows in the GUI.

Figure 2-6 Window menu



- **Show View**: Displays a view. This option is used to display a common panel that is closed, including the **Project Explorer**, **Resource Manager**, **Outline View**, and **Properties View**.
- **Reset Perspective**: Resets the current GUI layout.

2.1.5 Help

The **Help** menu provides some basic information and help document about the HiGVBuilder.

- **Help Content**: Opens the help PDF document.
- **About**: Displays basic information about the HiGVBuilder.

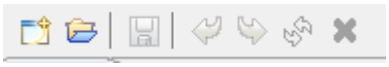
2.2 Toolbar

Buttons on the toolbar are divided by divider lines based on the button functions, and the divided bars can be dragged within a certain range.

2.2.1 HiGVBuilder Default Toolbar

The HiGVBuilder default toolbar contains buttons for common operations such as create, save, and search.

Figure 2-7 HiGVBuilder default toolbar



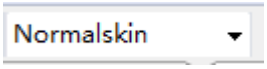
- : Displays the dialog box for creating a project or file.
- : Saves the edited content to a file.
- : Restores the content in the editor to the state before the latest operation is performed.
- : Restores the undo operation.
- : Refreshes the project list in the Project Explorer.
- : Deletes the selected content in the current editor.

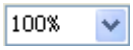
2.2.2 Graphics Editor Toolbar

The graphics editor toolbar facilitates graphics editing. It is displayed when the editor is opened, and hidden when the editor is closed. The buttons are described as follows:

Figure 2-8 Graphics editor toolbar

- (Align Left): Aligns the selected controls with the left edge of the last selected control.
- (Align Right): Aligns the selected controls with the right edge of the last selected control.
- (Align Vertical Center): Aligns the vertical central lines of all selected controls with that of the last selected control.

-  (Align Top): Aligns the selected controls with the top edge of the last selected control.
-  (Align Bottom): Aligns the selected controls with the bottom edge of the last selected control.
-  (Align Horizontal Center): Aligns the horizontal central lines of all selected controls with that of the last selected control.
-  (Align Width): Adjusts the spacing of all selected controls to even the horizontal spacing while keeping the leftmost and rightmost selected controls still.
-  (Align Height): Equalizes the spacing of all selected controls while keeping the topmost and bottommost selected controls still.
-  (Match Width): Equalizes the width of all selected controls based on the width of the last selected control.
-  (Match Height): Equalizes the height of all selected controls based on the height of the last selected control.
-  (Character String Width Adaptation): Adjusts the width of the current control (Label or Button) based on the length of the character string on the control. (The width of the current control is not changed when it is wider than the width of the character string.)
-  (Format Painter): Applies the formatting of a control to other controls.
- 

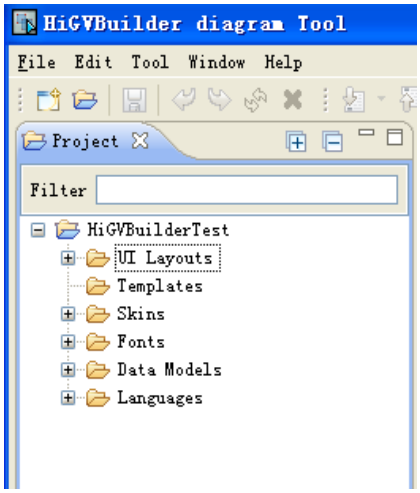
(Skin Switch): Switches the current displayed skin (skins for the normal state, activated state, highlighted state, disabled state, and mouse-down state).
- 

(Scale): The default value is 100%, that is, the original size is used.

2.3 Project Explorer

The Project Explorer manages project files. It displays project directories in the tree structure.

Figure 2-9 Project Explorer



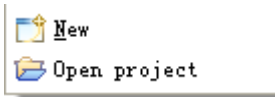
The upper left corner is the view identifier area that contains the view name and exit button. The upper right corner is the dedicated toolbar for the panel. Items in the panel are listed in the tree structure.

Buttons on the toolbar are described as follows:

- : Collapses all project nodes and displays only project names.
- : Expands all project nodes.
- : Minimize the panel. After this button is clicked, you can click to restore the panel position and size. All views have this button.
- : Maximize the panel. After this button is clicked, you can click to restore the panel position and size. All views have this button.

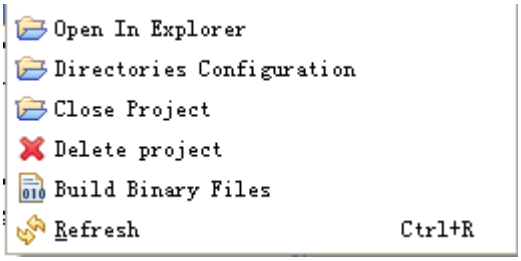
Project management is typically implemented by using shortcut menus. The following describes the functions of shortcut menus.

Figure 2-10 Shortcut menu when no project is selected in the panel



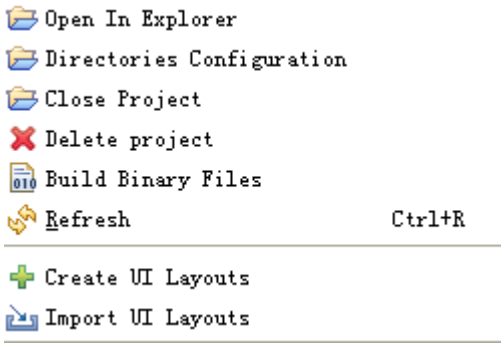
- **New**: Creates a project and drawing.
- **Open Project**: Opens a project.

Figure 2-11 Shortcut menu when a project is selected in the panel



- **Open In Explorer**: Opens the directory for storing the project in the Explorer.
- **Directories Configuration**: Sets the resource directories configured before in the project.
- **Close Project**: Closes the project being edited currently.
- **Delete project**: Removes the selected project.
- **Build Binary Files**: Generates the .bin files required by the GUI by using the XML files.
- **Refresh**: Refreshes the project.

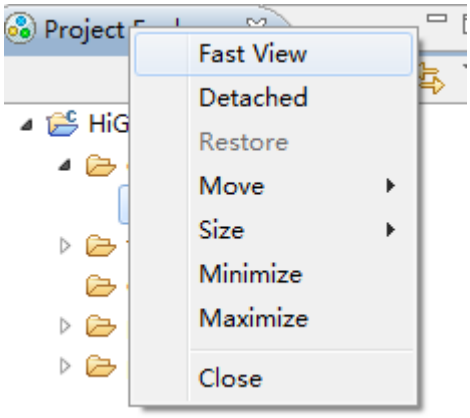
Figure 2-12 Shortcut menu when a file in a project is selected



When different files are selected, the options for creating and importing related UI layouts are added.

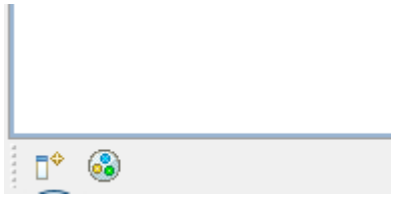
Right-click the project management view identifier area. The view management menu is displayed.

Figure 2-13 View management menu



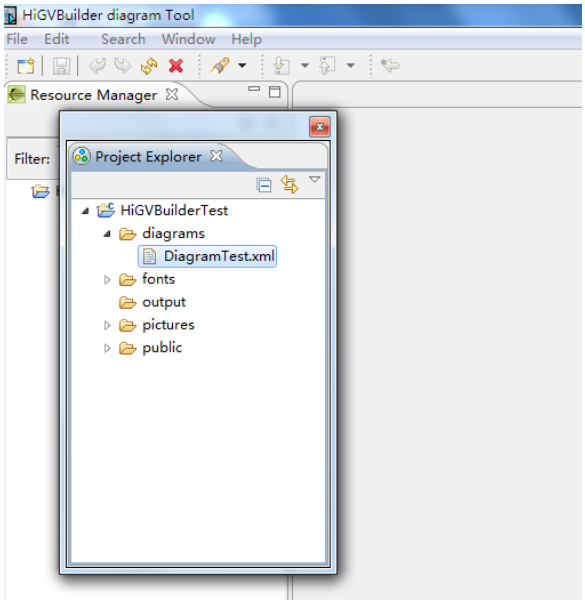
- **Fast View**: Displays the view in the fastboot area (left part of the status bar).

Figure 2-14 Fast view



- **Detached**: Detaches the view from the main window.

Figure 2-15 Detached view



- **Restore**: Restores the current view to the default state.

- **Move (View/Tab Group)**: Moves the view between tab pages in the main window or view window.
- **Size**: Adjusts the view size. This option is available only when the view is not in detached state.
- **Minimize**: Minimizes the view and displays the view icon in the nearest window edge bar.
- **Maximize**: Expands the view to fit into the window and minimizes other views and editing areas.
- **Close**: Closes the view.

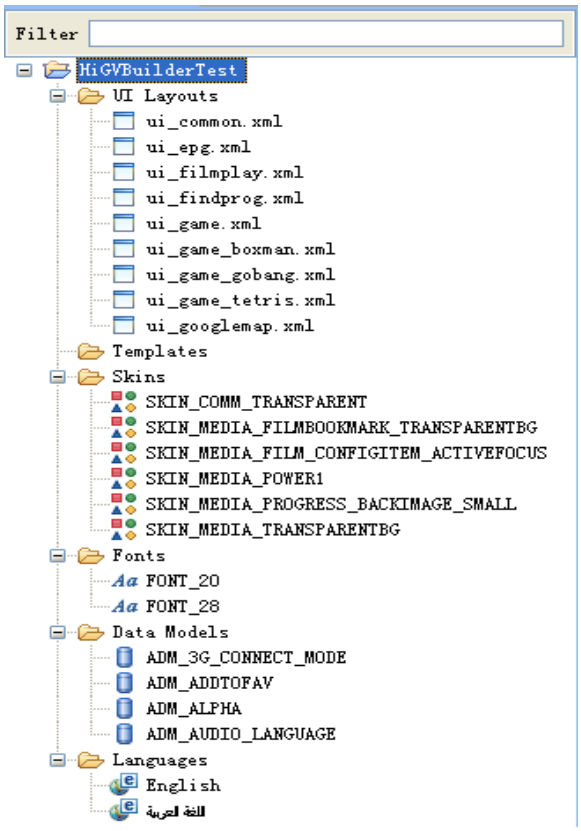
NOTE

The management menus for other views except the GUI editor view are the same as that for the Project Explorer.

2.4 Resource Manager

The resource manager manages all drawings and public resources in a project. See Figure 2-16.

Figure 2-16 Resource Manager



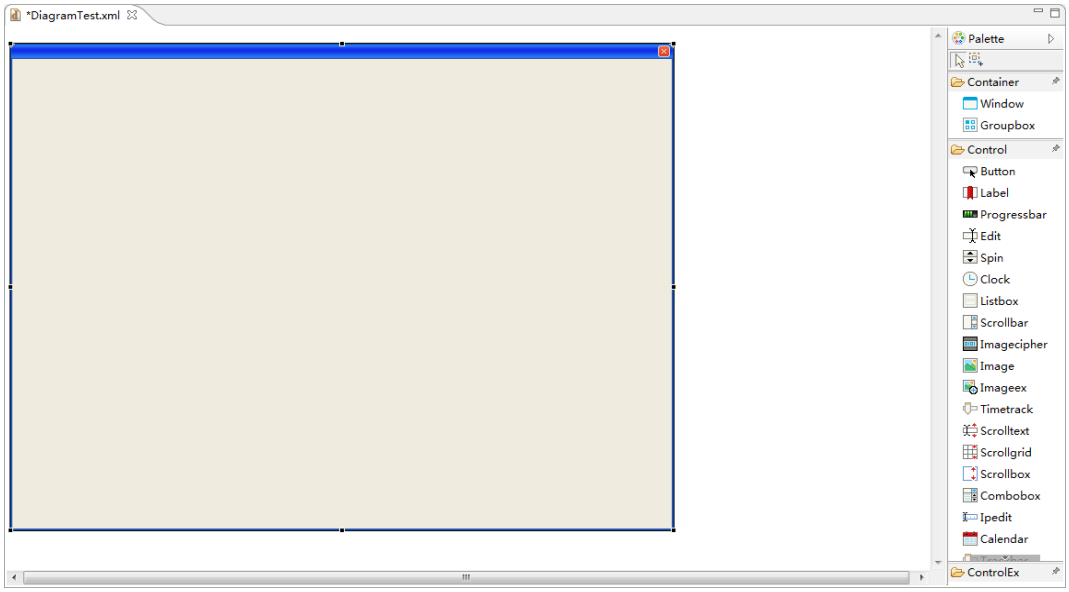
Different from the Project Explorer, the Resource Manager lists all drawings and public resources for a project by category (not corresponding to files), and provides related management functions (which are described in the public resource management section).

This view allows you to filter resources by name. When you enter a keyword in the **Filter** textbox, the drawings and resources whose names do not contain the keyword are hidden (the directory nodes are not hidden).

2.5 GUI Editor View

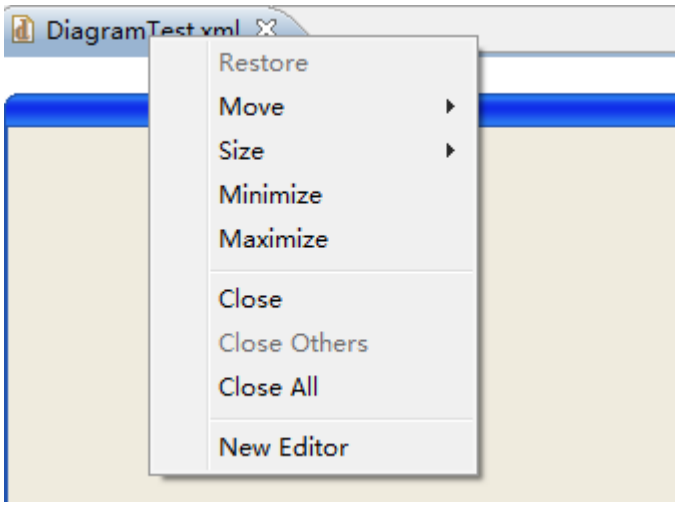
The GUI editor view is the main editing area for drawing all GUIs. All containers or controls can be edited only after they are put into the editor.

Figure 2-17 GUI editor view



The view management menu of the GUI editing area is different from that of other views. See Figure 2-18.

Figure 2-18 Management menu of the GUI editor view



- **Close Others**: Closes all other editors that are opened.
- **Close All**: Closes all editors.
- **New Editor**: Opens a new editor. The content in the new editor is the same as that in the current editor.

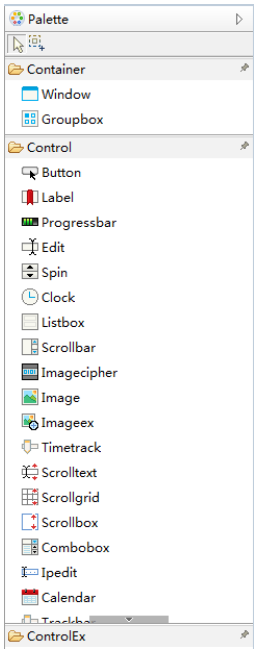
NOTE

For details about other menu items, see the description of the management menu for the Project Explorer.

2.5.1 Tool Zone

The tool zone clings to the frame of the graphics editing area. It is used to store various containers and controls.

Figure 2-19 Graphics tools



 in the identifier area is used to expand or collapse the tool zone.

There are two buttons on the tool bar:

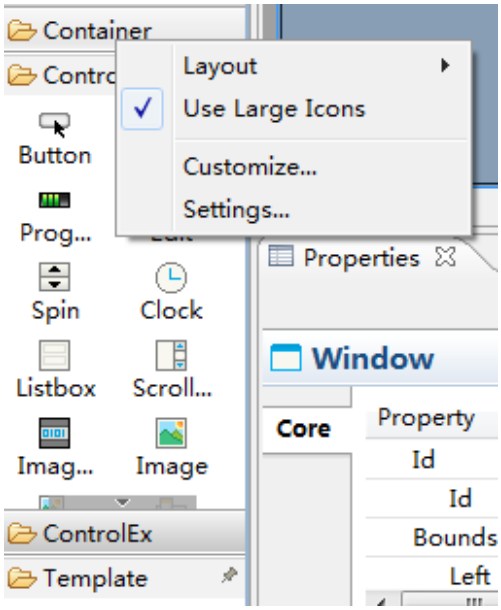
-  : radio button (default). When this button is clicked, you can click a control to select it.
-  : checkbox button. When this button is clicked, you can select multiple controls at a time by selecting checkboxes.

Both the container control area and non-container control area can be expanded and collapsed.

You can scroll the list of controls in the control area by clicking and  .

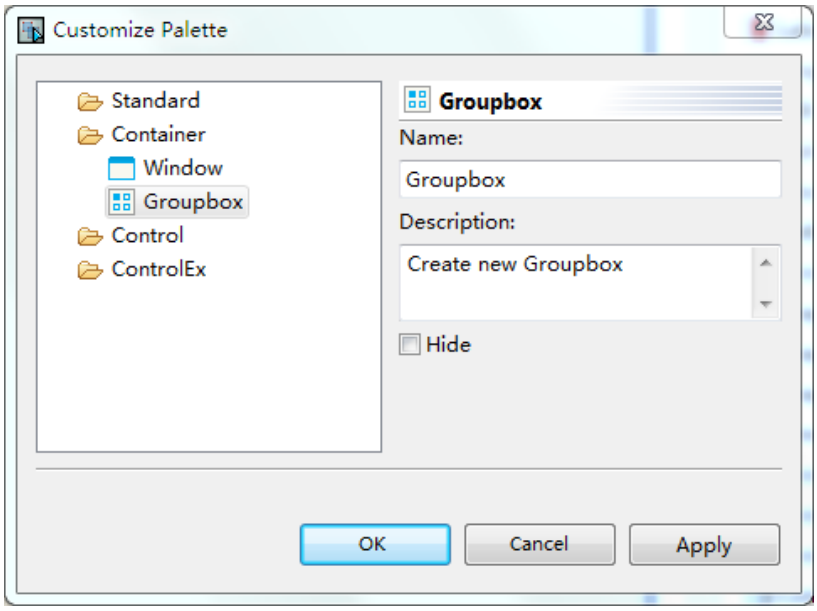
Figure 2-20 shows the shortcut menu of the tool zone.

Figure 2-20 Shortcut menu of the tool zone



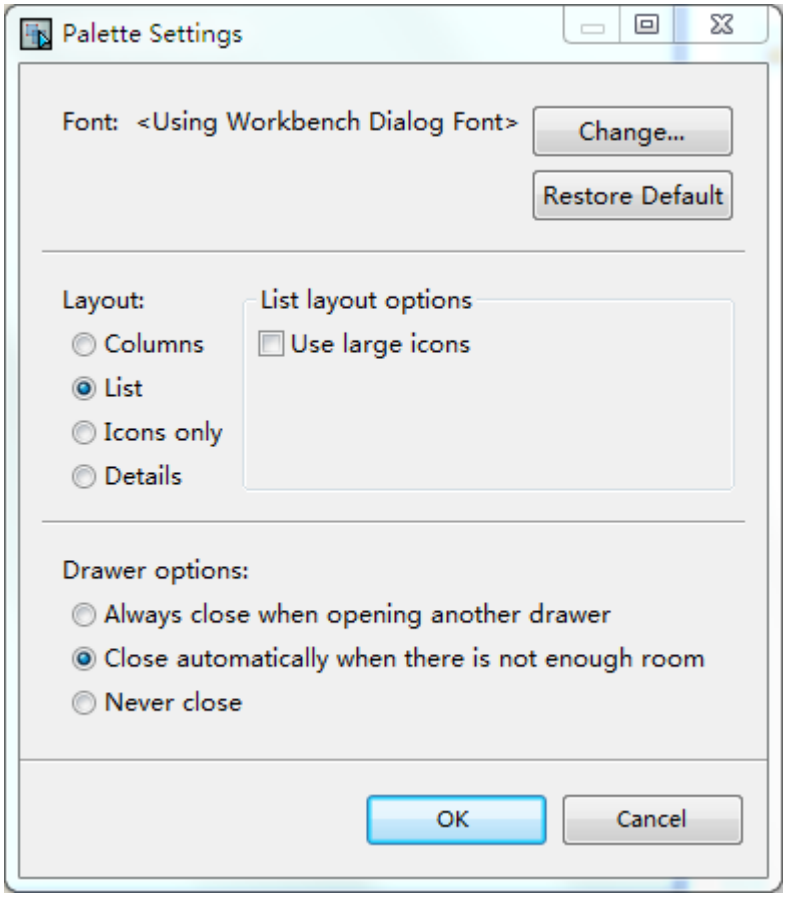
- **Layout**: Selects the layout of controls in the tool zone. The HiGVBuilder provides four layout modes: **Columns**, **List** (default), **Icons Only**, and **Details**.
- **User Large Icons**: Changes the icons of four buttons such as the radio button to large icons. It is optional.
- **Customize**: Displays the customization dialog box for setting the control names and describing or hiding the controls.

Figure 2-21 Customization dialog box



- **Settings**: Displays the dialog box for setting detailed parameters of the tool zone.

Figure 2-22 Settings dialog box



- Pinned: After the pin is clicked, the small area for storing controls by category is not overlaid with other areas. It is optional.

The shortcut menu by clicking the tool zone identifier area is described as follows:

- **Resize**: Resets the size.
- **Dock On**: There are two options: **Left** and **Right** (default).

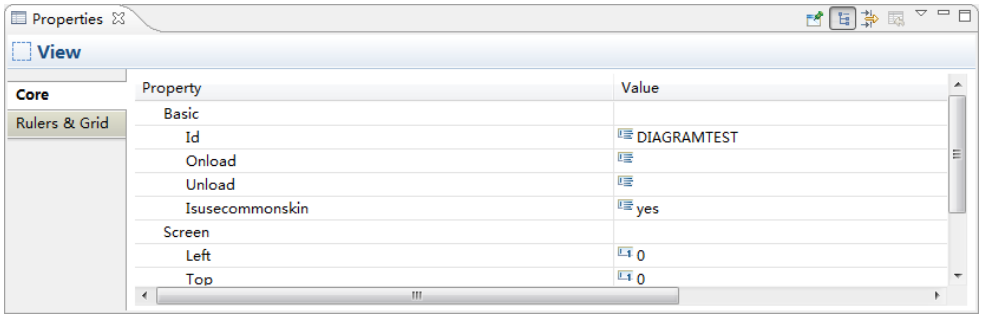
NOTE

You can set the tool zone position by using the **Dock On** option in the shortcut menu or directly dragging the tool zone.

2.5.2 Properties View

The Properties view displays control properties and provides a view for editing properties. The properties are displayed in two columns: **Property** and **Value**.

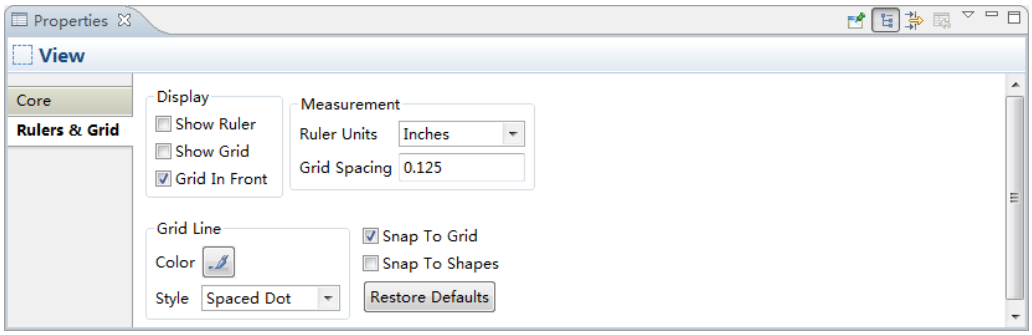
Figure 2-23 Properties view



The following describes the buttons of the tool bar in the upper right corner:

- : Displays the status bar of the control property area.
- : Displays properties in the tree structure (selected by default).
- : Displays advanced properties.

Figure 2-24 Rulers & Grid



When you click any blank area in the editor, the **Rulers & Grid** option is displayed in the property area. You can select the option to display the ruler and grid setting panel.

- **Display**: display area
  - **Show Ruler**: Displays the ruler.
  - **Show Grid**: Displays the grid.
  - **Grid In Front**: Displays the grid at the front.
- **Measurement**: measurement unit

  Ruler Units:
  - Centimeters (default)
  - Pixels
  - Inches

  Grid Spacing
- Grid Line
- Color (color of the grid line)
- Style (grid line type)

  - Solid
  - Dash
  - Dot
  - Dash Dot
  - Dash Dot Dot
  - Spaced Dot
- Others
  - Snap To Grid
  - Snap To Shapes
  - Restore Defaults

2.5.3 Outline View

The Outline view displays the containers and controls in the graphics editing area in thumbnails or tree structure.

Figure 2-25 Thumbnail

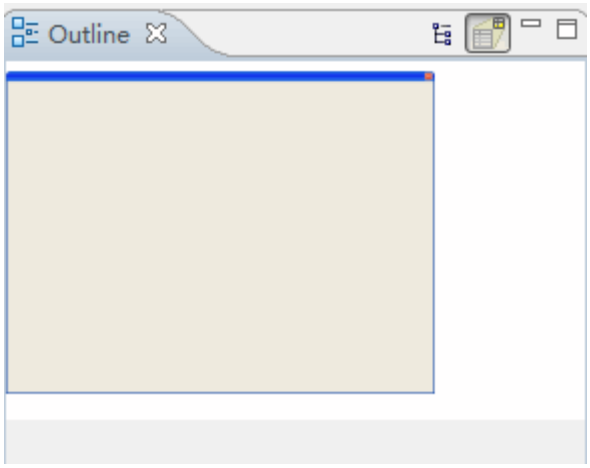
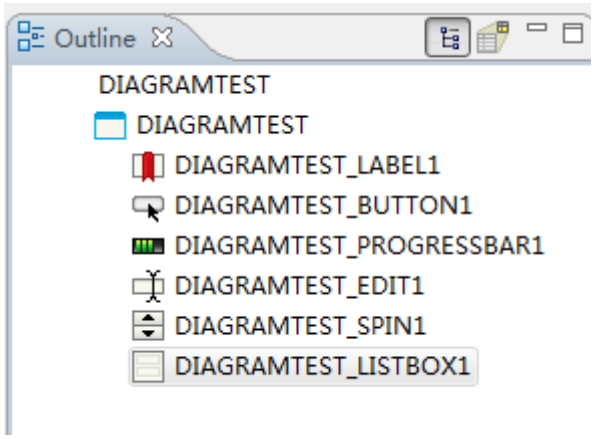
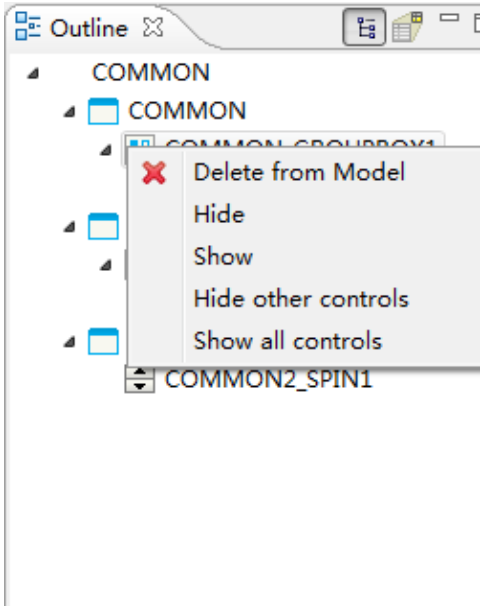


Figure 2-26 Tree outline



- : Displays in the tree structure. Control IDs are used as the nodes of the tree.
- : Displays in graphics, that is, the thumbnail.

Figure 2-27 Shortcut menu



- **Delete from Model**: Deletes the currently selected control in the GUI.
- **Hide**: Hides the currently selected control.
- **Show**: Displays the currently selected control.
- **Hide other controls**: Hides all other container controls except the selected control.
- **Show all controls**: Displays all controls.

NOTE

The hide and display functions here are useful for complex UI components (such as the window and Groupbox) which contain many child controls. These controls may overlap. The controls that do not need to be edited currently can be hidden to facilitate the editing of the current selected control.

2.6 Aligning Controls in A Container Control

When a container control contains child controls, the following alignment options are available:

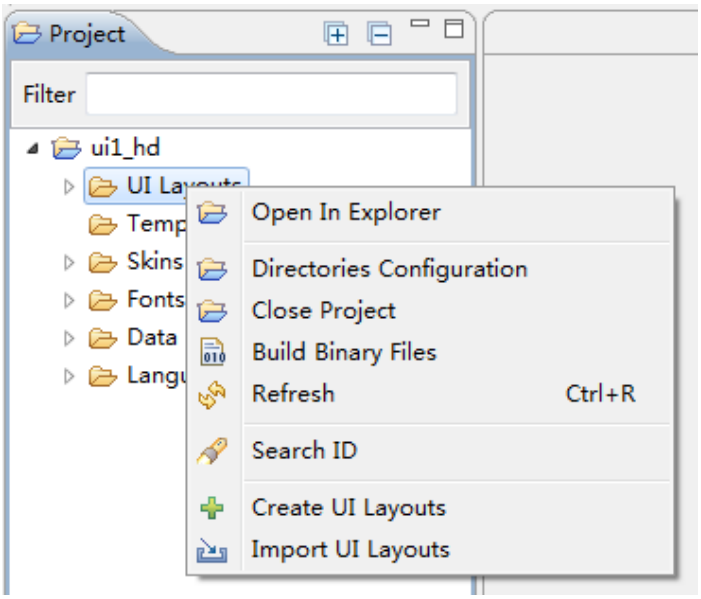
- Align Left: Aligns all selected controls with the left edge of the container control.
- Align Right: Aligns all selected controls with the right edge of the container control.
- Align Vertical Center: Aligns the vertical central lines of all selected controls with that of the container control.
- Align Top: Aligns all selected controls with the top edge of the container control.
- Align Bottom: Aligns all selected controls with the bottom edge of the container control.
- Align Horizontal Center: Aligns the horizontal central lines of all selected controls with that of the container control.

2.7 Searching For a UI Control ID

You can search for the UI on which a control is displayed based on the control ID. After search, the UI is automatically displayed and the control with the searched ID is selected. To search for a control ID, perform the following steps:

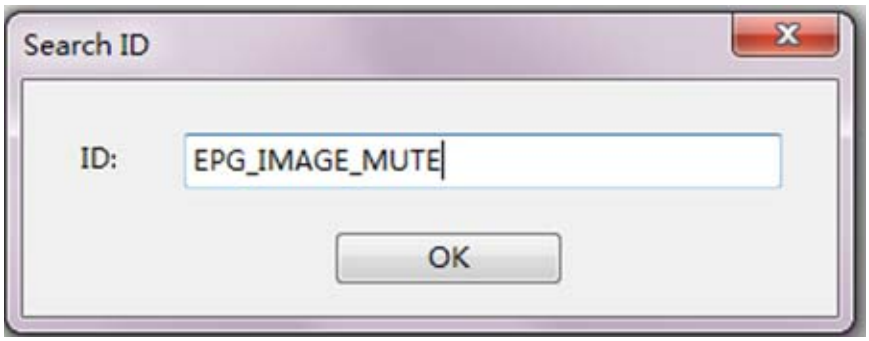
Step 1 Right-click **UI Layouts** and select **Search ID** from the shortcut menu as shown in Figure 2-28.

Figure 2-28 Selecting Search ID



Step 2 On the displayed **Search ID** dialog box, enter a control ID in the **ID** text box and click **OK**, as shown in Figure 2-29.

Figure 2-29 Entering an ID



----End

2.8 Saving the UI XML Comments

To save the comments of the UI XML file that conform to the standards, perform the following steps:

Step 1 Open the UI XML source file in the Notepad.

Step 2 Add the comment **<!--window epg -->** before **<window id="EPG"** and save the file, as shown in Figure 2-30.

Figure 2-30 Comment

```
<?xml version="1.0" encoding="utf-8"?>
<view id="EPG"
      isusecommonskin="no">
  <!--window epg -->
  <window id="EPG"
          width="1280"
          height="720"
          isrelease="yes"
          normalskin="common_transparent_skin"
          ontimer="EPGWindowOnTimer"
          widgetposmirror="yes"
          widgetinteriormirror="yes"
          onkeydown="EPGWindowOnKeyDown"
          opacity="255"
          winlevel="5"
          onevent="EPGWindowOnEvent"
          onshow="EPGWindowOnShow"
          onrefresh="EPGWindowOnRefresh"
          onhide="EPGWindowOnHide">
```

----End

CAUTION

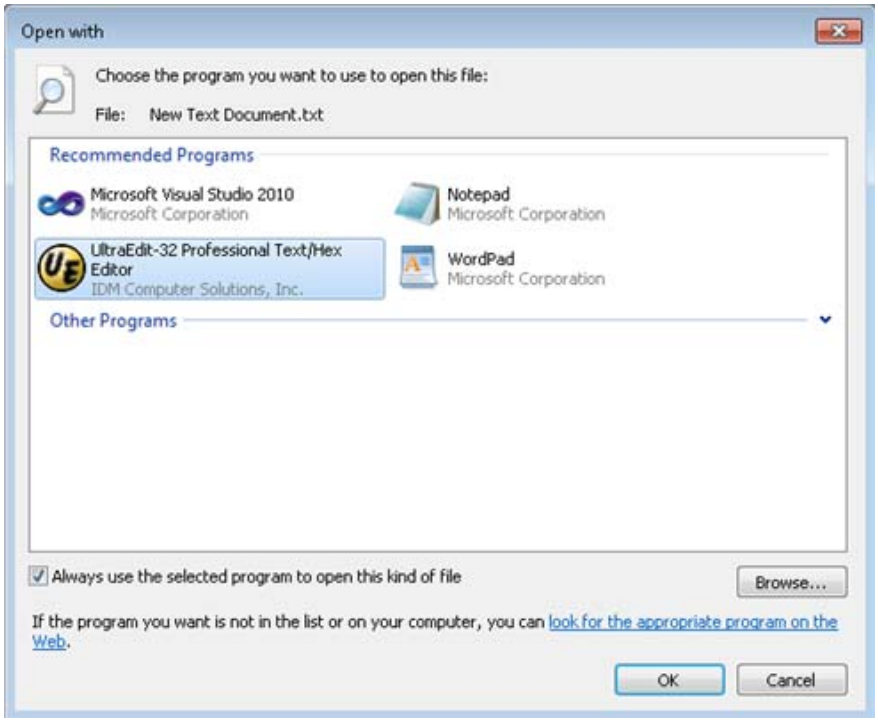
The comments must be written in the format of **<!--*window epg* -->** and before the left angle brackets of a control.

2.9 Opening a UI XML File with the System Editor

To open the UI XML file by using the system default editor, perform the following steps:

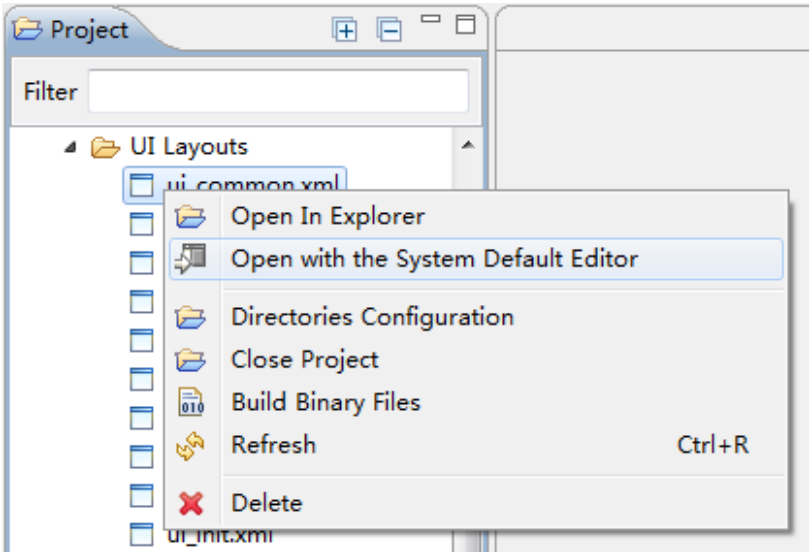
Step 1 Right-click the UI XML source file, select **UltraEdit-32 Professional Text/Hex Editor** and **Always use the selected program to open this kind of file**, and click **OK**, as shown in Figure 2-31.

Figure 2-31 Specifying a program to open the XML file



Step 2 Under the **UI Layouts** directory, select the UI XML source file and right-click the file, select **Open with the System Default Editor** as shown in Figure 2-32.

Figure 2-32 Opening the UI XML file with the System Default Editor



----End

3 Application Reference

3.1 Overview

Project management, public resource management, and GUI editing must be performed in the graphics editing perspective.

3.1.1 Creating a Project

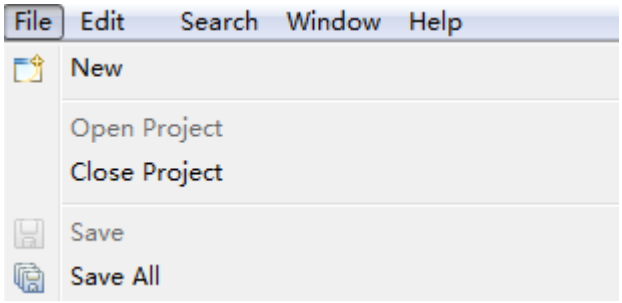
You can create a project by using any of the following three methods:

- Right-click the project management area, and choose **New** from the shortcut menu, as shown in Figure 3-1.

Figure 3-1 Creating a project (method 1)

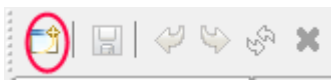
- Choose **File** > **New**, as shown in Figure 3-2.

Figure 3-2 Creating a project (method 2)



- Click the control for creating a project on the toolbar, as shown in Figure 3-3.

Figure 3-3 Creating a project (method 3)



The dialog box for creating a project is displayed when any of the preceding operations is performed.

CAUTION

The project name cannot be duplicated or contain special characters.

3.1.2 Importing a Project

You can import an existing project to edit it or perform other operations by using either of the following two methods:

- Choose **Open Project** from the shortcut menu.
- Choose **File** > **Open Project**.

3.1.3 Closing a Project

Projects that are not used currently can be closed to reduce memory usage. After a project is closed, the project resources cannot be used.

You can close a project by using either of the following methods:

- Select the project, and choose **Close Project** from the shortcut menu.
- Select the project, and choose **File** > **Close Project**.

The icon of the closed project is  .

3.1.4 Deleting a Project

You can delete a project permanently by using either of the following two methods. After the operation, the project name is not displayed any more.

- Select the project, and choose **Delete Project** from the shortcut menu.
- Select the project, and choose **File** > **Delete Project**.

3.1.5 Opening a Project

You can open a closed project to edit it by using either of the following methods:

- Select the project, and choose **Open Project** from the shortcut menu.
- Select the project, and choose **File** > **Open Project**.

3.1.6 Refreshing Project Files

When you refresh a project, all files of the project are reloaded from the local working directory. This operation can be used to quickly refresh the project after some configurations or files of the project are modified.

You can refresh a project by using either of the following methods:

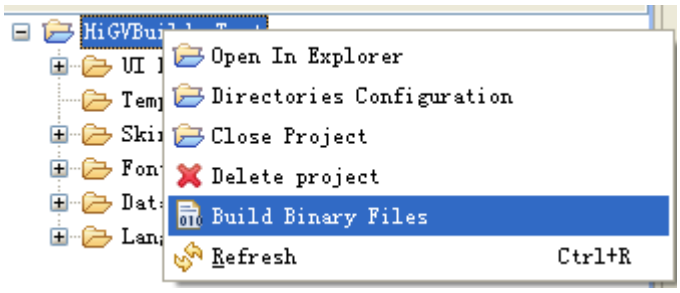
- Select the project, and choose **Refresh** from the shortcut menu.
- Select the project, and choose **File** > **Refresh**.

After being refreshed, the entire project is reloaded.

3.1.7 Generating HiGV Binary Files

You can generate the binary UI files required by the HiGV by choosing **Build Binary Files** from the shortcut menu of the project management area, as shown in Figure 3-4.

Figure 3-4 Generating HiGV binary files



The generated default files such as .bin files are stored in the **Output** directory of the current project.

NOTE

The HiGVBuilder copies only the generated .bin files, language files, and necessary code files to the **Output** directory. The generated files include **XX.lang**, **higv.bin**, **higv_cextfile.c**, **higv_cextfile.h**, **higv_language.h**, and **XXX.c** (if a callback event is set for the UI description files, a corresponding .c file will be generated).

CAUTION

Each time you click **Build Binary Files**, a new function header related to the GUI is generated. You need to back up the .c files before modifying them.

If no error occurs, you can view all generated files in the **Output** directory after the generation.

3.1.8 Opening a Recent Project

Files that are opened recently in the HiGVBuilder are listed at the bottom of the **File** menu. You can directly open the recent files from the **File** menu. A maximum of four recent files are listed. The first one is the latest.

3.2 Project Resources and Management

3.2.1 Introduction

The HiGV GUI uses five resources:

- (drawing): The GUI drawings are edited by the user in the editor.

- (skin): The skin determines the appearance of controls. It consists of nine parts called nine squares and can be created by using pure color blocks or images.
- (font): The font determines the type, size, and style of characters displayed on controls.
- (data model): The data model describes the data models (information such as the database field) used by controls. It is important for binding data when the GUI code is compiled in the late phase.
- (language): The HiGV GUI language consists of multiple local character strings.

You can manage the preceding project resources by using the Resource Manager.

CAUTION

The skins, fonts, data models, and multi-language character strings are stored in the corresponding XML files regardless of their quantity.

3.2.2 Drawing

You can manage drawings in the Project Explorer. To create a drawing, right-click the drawing directory of the corresponding project and choose **Create UI Layouts**. For details about the follow-up operations, see section 1.3.5 "Create a GUI."

To delete a drawing, right-click the drawing file and choose **Delete**.

The system displays a dialog box to confirm the delete operation.

NOTE

In the Resource Manager, you can select multiple resources (even the resource types are different, for example, drawings and fonts) or projects by pressing **Ctrl** and delete these resources at a time by choosing the **Delete** option. When you delete a project, a dialog box is displayed asking you whether to delete all files in the project directory. The classification directories for various resources are not deleted.

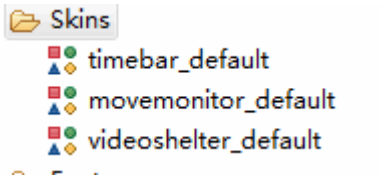
3.2.3 Skin

3.2.3.1 Introduction

The skin determines the appearance of all controls (including the window container). In the HiGV, the skin consists of nine parts (left top, top, right top, left, middle, right, bottom left, bottom, right bottom) of images (or colors), which is called the nine square skin. The nine square skin allows the image resources to be used repeatedly. The same set of image resources can be combined to obtain skins with the same style but different sizes or with different styles.

Skins created by the user are listed in the **Skins** directory of each project in the Resource Manager, as shown in Figure 3-5.

Figure 3-5 Skins directory



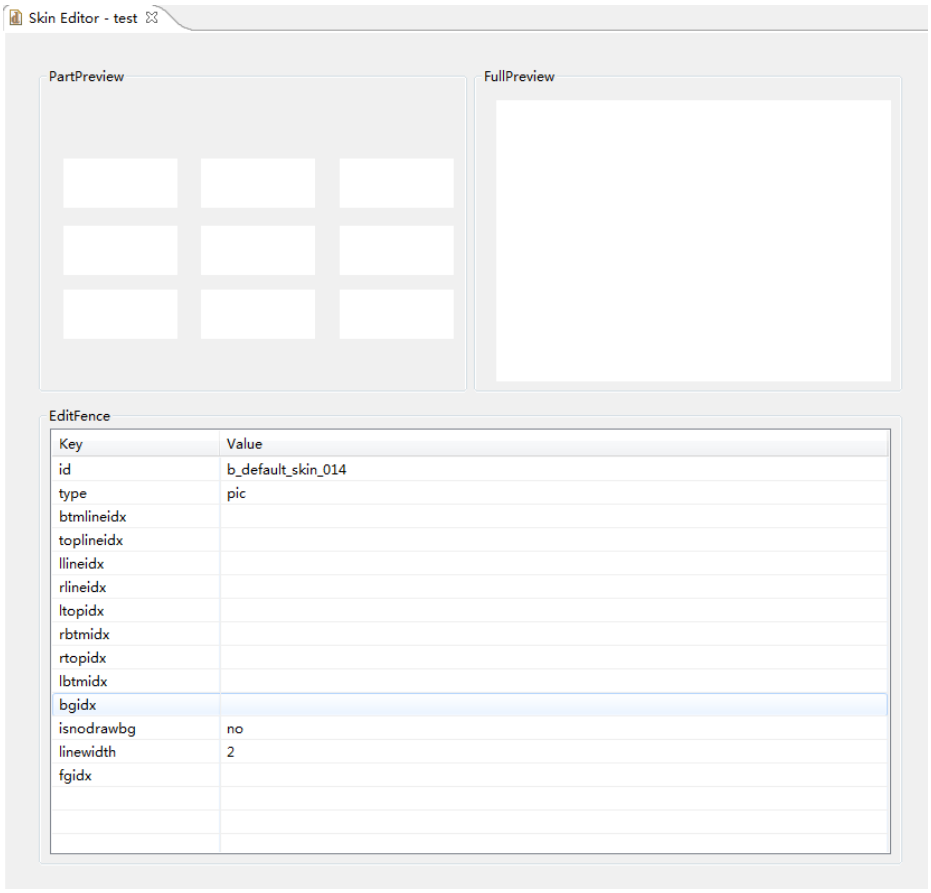
NOTE

For more details about the skin resources, see the *HiGV Development Guide*.

3.2.3.2 Creating a Skin

Right-click the **Skins** directory of the current project, choose **Create Skins**, enter the resource ID, and select the XML file for storing the resource ID. The editing GUI for creating skins is displayed.

Figure 3-6 Skin editor



The skin editing GUI is divided into three zones:

- **PartPreview**: Previews each nine-square image.
- **FullPreview**: Previews the full image after the nine squares are combined.
- **EditFence**: Edits the skin.

- **id**: unique skin ID. A default ID is generated automatically based on the project name.
- **linewidth**: filling color width
- **isnodrawbg**: whether to fill the background color. **yes** indicates that the background color is not filled, and **no** indicates that the background color is filled.
- **type**: skin resource type. **pic** indicates the image type, and **color** indicates the color type.
- **llineidx** to **bgidx**: color values or file paths for setting the skin. If the current skin type is **color**, only color values (8-bit or 10-bit color character strings starting with 0x) are allowed in these fields, and clicking the **Browse** button of the fields opens the color selector. If the current skin type is **pic**, only file paths are allowed in these fields, and clicking the **Browse** button of the fields opens the file selector.

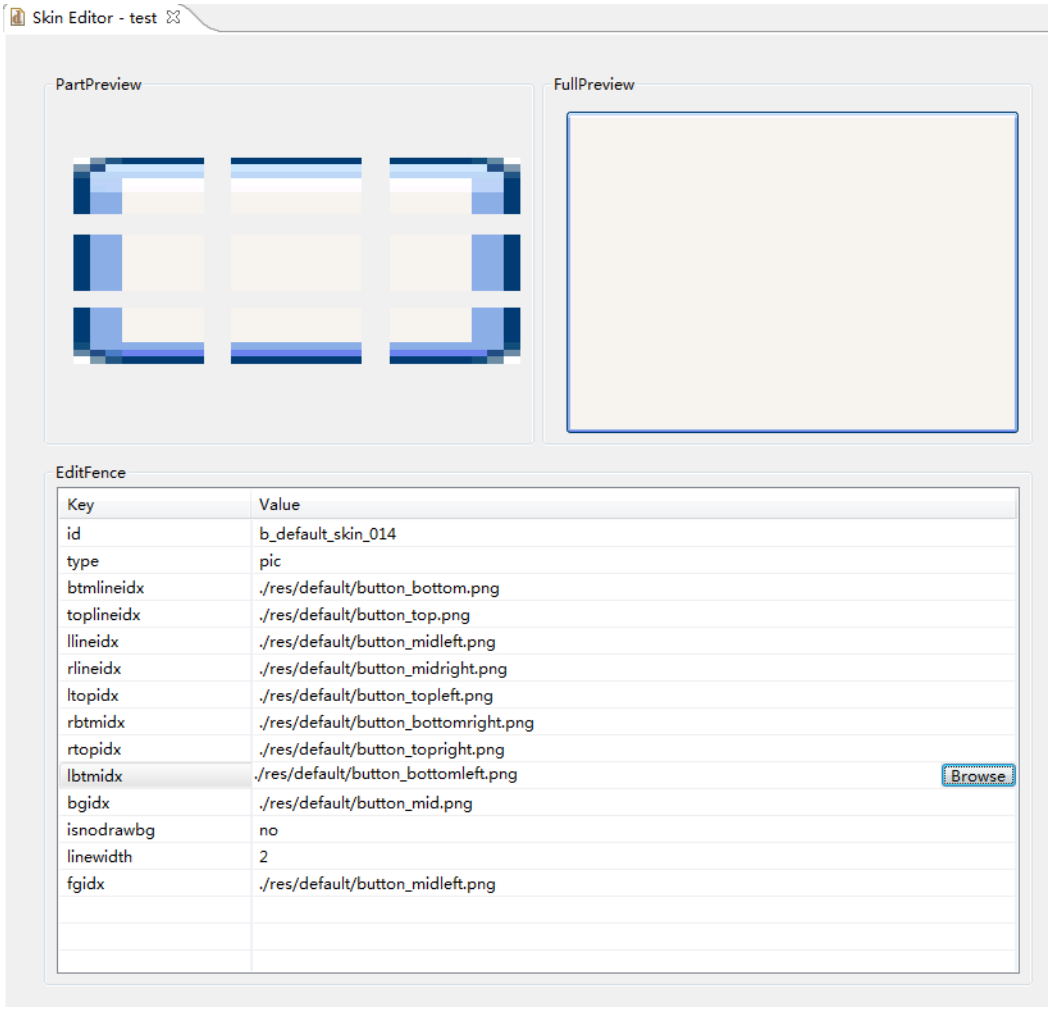
NOTE

- If the skin type is **pic**, only images in the **picture** directory of the current project can be selected.
- If the entire image is displayed, only **bgidx** needs to be configured for the nine square skin.

- **fgidx**: frontground color (only color values for the character string are allowed). The character string in the selected status can be displayed in different colors by changing the **fgidx** value of **activeskin**.

After filling the preceding fields, you can click the save button on the toolbar to add the new skin resource to the resource library.

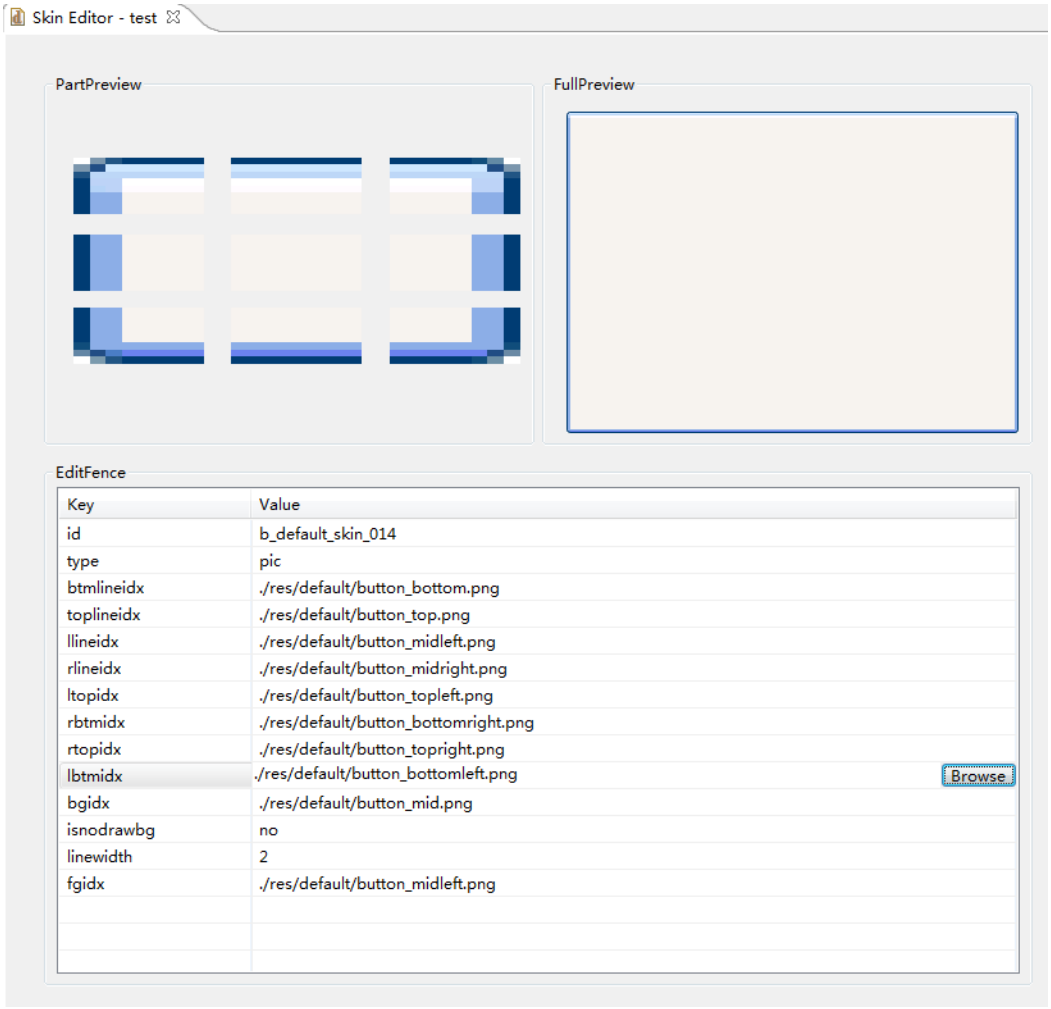
Figure 3-7 Skin editor after the skin is created



3.2.3.3 Modifying a Skin

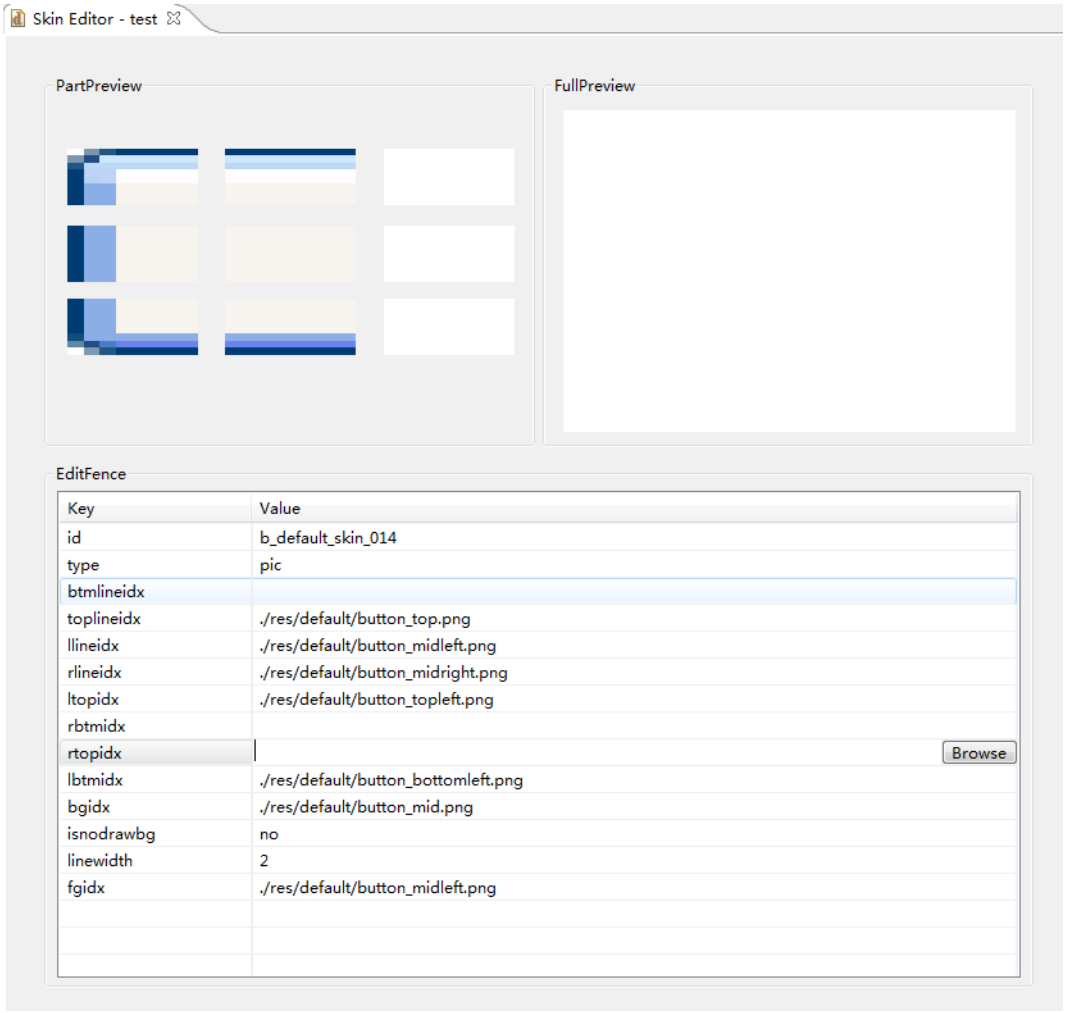
To modify a skin, double-click the skin icon in the skin directory.

Figure 3-8 Modifying a skin



After modification, click the save button on the toolbar to save the modification.

Figure 3-9 Saving the modification



3.2.3.4 Deleting a Skin

To delete a skin, right-click the skin file in the Resource Manager, and choose **Delete**.

CAUTION

If the skin is used, it cannot be displayed after being deleted.

3.2.3.5 Importing a Skin

To import a skin, right-click the **Skins** directory, choose **Import Skin**, and select the skin XML file to be imported in the displayed dialog box.

CAUTION

When a skin is imported, only the skin XML file but not the skin image is imported. You need to import all the images used in the XML file to the **pictures** directory in the Project Explorer.

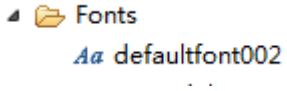
3.2.4 Font

3.2.4.1 Introduction

The font determines the style of texts displayed on text controls (such as the Button and Label). Different from common fonts, the HiGV font consists of the font source file and parameter configurations.

The **Fonts** directory of each project in the Resource Manager lists the defined fonts in the corresponding project, as shown in Figure 3-10.

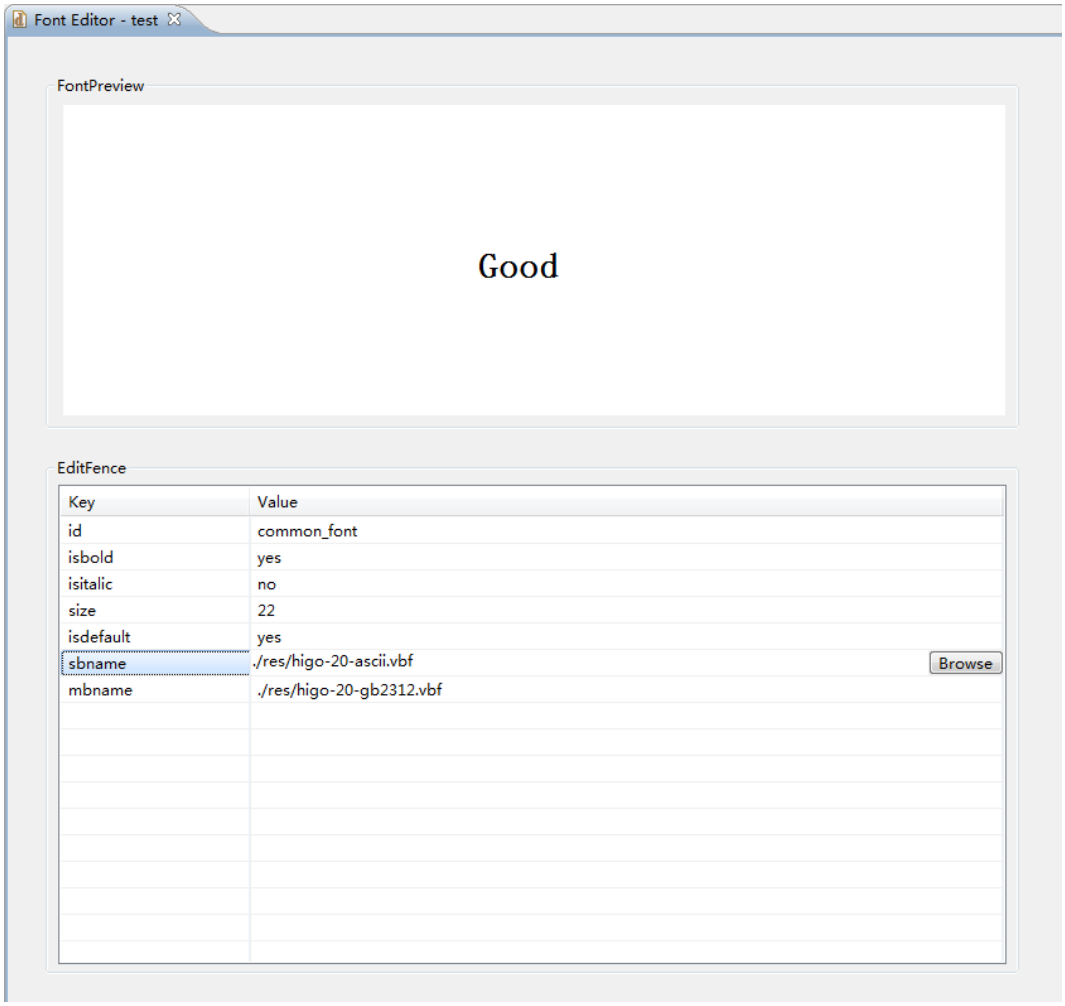
Figure 3-10 Fonts directory



3.2.4.2 Creating a Font

Right-click the **Fonts** directory, choose **Create Fonts**, enter the resource ID, and select the XML file for storing the resource ID. The font editor is opened.

Figure 3-11 Creating a font

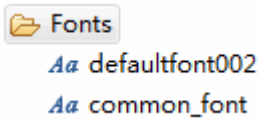


- **id**: unique font ID
- **bold**: whether the font is bold
- **italic**: whether the font is italic
- **size**: default font size, configurable
- **isdefault**: whether the font is the default font
- **sbname**: single-byte font file (for displaying fonts of characters defined by the ASCII code). You can enter the file path or select the file by clicking the **Browse** button.
- **mbname**: multi-byte font file (for displaying fonts of characters defined by non-ASCII codes). You can enter the file path or select the file by clicking the **Browse** button.

After setting the preceding parameters, you can preview the font effect in the **FontPreview** area. The previewed font is drawn by using the system default font by default. However, if the font is in TTF format, the character in the **FontPreview** area is drawn by using the selected TTF font.

After the preceding parameters are set, click the save button on the toolbar. Then you can find the created font in the Resource Manager, as shown in Figure 3-12.

Figure 3-12 Font list



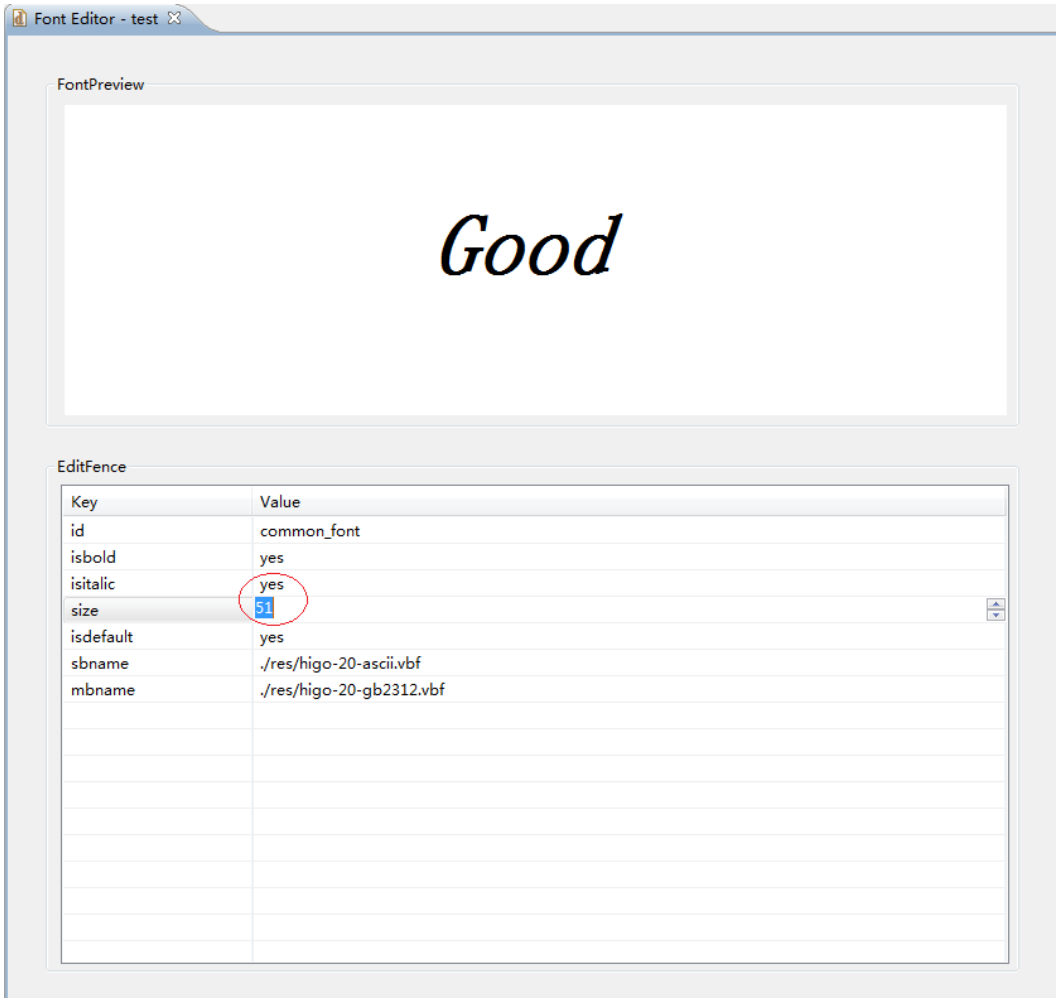
3.2.4.3 Modifying a Font

To modify a font, double-click the font in the **Fonts** directory.

The procedure for modifying a font is the same as that for creating a font, except that the font ID cannot be changed.

After modification, click the save button on the toolbar.

Figure 3-13 Modifying a font



3.2.4.4 Deleting a Font

To delete a font, right-click the font in the **Fonts** directory, and choose **Delete**.

CAUTION

Texts cannot be displayed properly after the corresponding font is deleted.

3.2.4.5 Importing a Font

To import a font, right-click the **Fonts** directory, choose **Import Font**, and select the font XML file to be imported in the displayed dialog box.

CAUTION

Only the .xml font files can be imported. When a font is imported, the font source file is not imported at the same time. You need to import the font source files to the **Fonts** directory in the Project Explorer first.

3.2.5 Language

3.2.5.1 Introduction

The language of the character strings displayed on controls can be changed. Table 3-1 lists the supported languages (In the HiGV, the language code is defined based on the ISO639-2 standard, and the country and region code abbreviations are defined based on the ISO3166-1 standard.):

Table 3-1 Supported languages

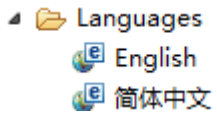
Identifier	Language	Description
cn	Simplified Chinese	-
zh	Simplified Chinese	Equivalent to cn
zh_CN	Simplified Chinese	Equivalent to cn
zh_TW	Traditional Chinese (Taiwan)	-
zh_HK	Traditional Chinese (HongKong)	-
ar	Arabic	-
cs	Czech	-
da	Danish	-
de	German (Germany)	-
de_AT	German (Austria)	-
de_CH	German (Switzerland)	-
el	Greek	-

Identifier	Language	Description
en	English	Equivalent to en_US
en_US	English (US)	Equivalent to en
en_CA	English (Canada)	-
en_GB	English (UK)	-
es	Spanish	-
fa	Persian	-
fi	Finnish	-
fr	French (France)	-
fr_CA	French (Canada)	-
fr_CH	French (Switzerland)	-
it	Italian (Italy)	-
it_CH	Italian (Switzerland)	-
he	Hebrew	-
ja	Japanese	-
ko	Korean	-
ko_KR	Korean (South Korea)	-
nl	Dutch (the Netherlands)	-
nl_BE	Dutch (Belgium)	-
pt	Portuguese (Portugal)	-
pt_BR	Portuguese (Brazil)	-
ru	Russian	-
sv	Swedish	-
th	Thai	-
tr	Turkish	-
In	Indonesian	

In the **Language.xml** resource file, items in the **Identifier** column represent specific languages. For example, simplified Chinese can be represented by zh or zh_CN, and English can be represented by en, en_US, or en_GB based on the country.

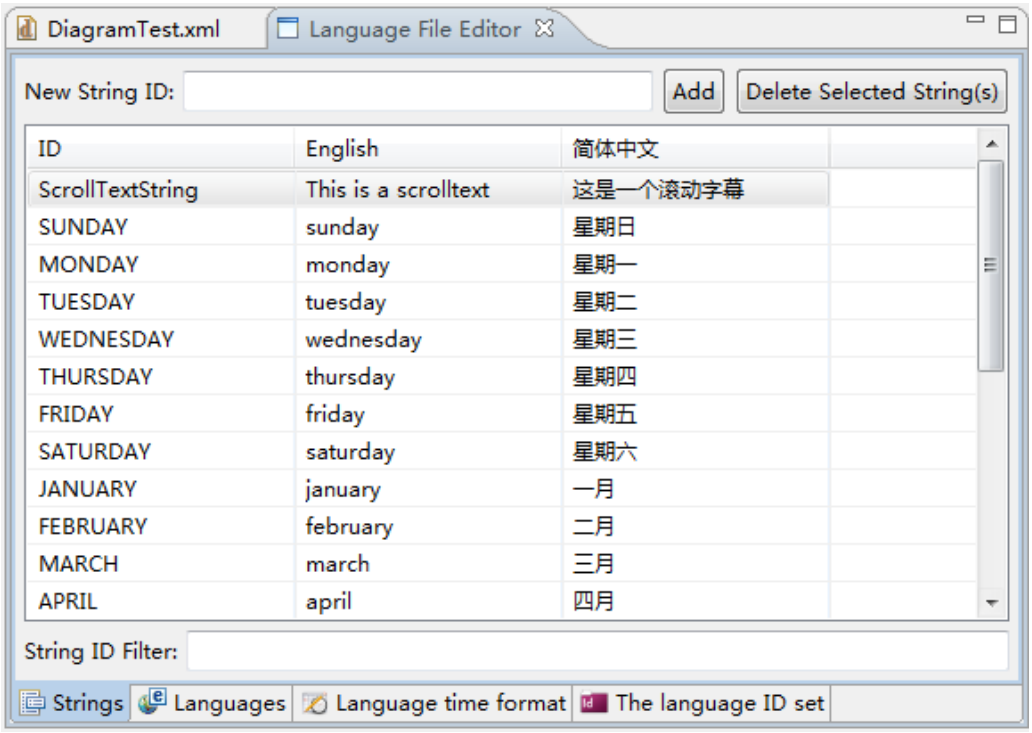
The **Languages** directory of each project in the Resource Manager lists all defined languages.

Figure 3-14 Language list



You can click the icon of any language to open the language editor. In the language editor, you can manage multi-language character strings as well as multiple languages. See Figure 3-15.

Figure 3-15 Multi-language character string editor



After editing, you can click  on the toolbar to save the language configurations.

3.2.5.2 Managing Multi-Language Character Strings

To edit character strings, click the **Strings** tab at the bottom of the language editor. All defined language IDs and character strings are listed in the list box of the language editor. When a project is created, the Chinese and English are configured automatically, and therefore the Chinese and English character strings are displayed.

The ID column cannot be modified. You can click a cell in the specific language column to edit the character string with the specific ID in a specific language.

You can set **New String ID** in the upper area of the language editor to add a new character string ID. Enter a new ID, press **Enter** or click **Add**. Then the new ID is added to the list and the character string can be edited.

To delete a language character string, right-click the row to be deleted, and choose Delete Selected String(s). You can press Ctrl to select and delete multiple language character strings at the same time.

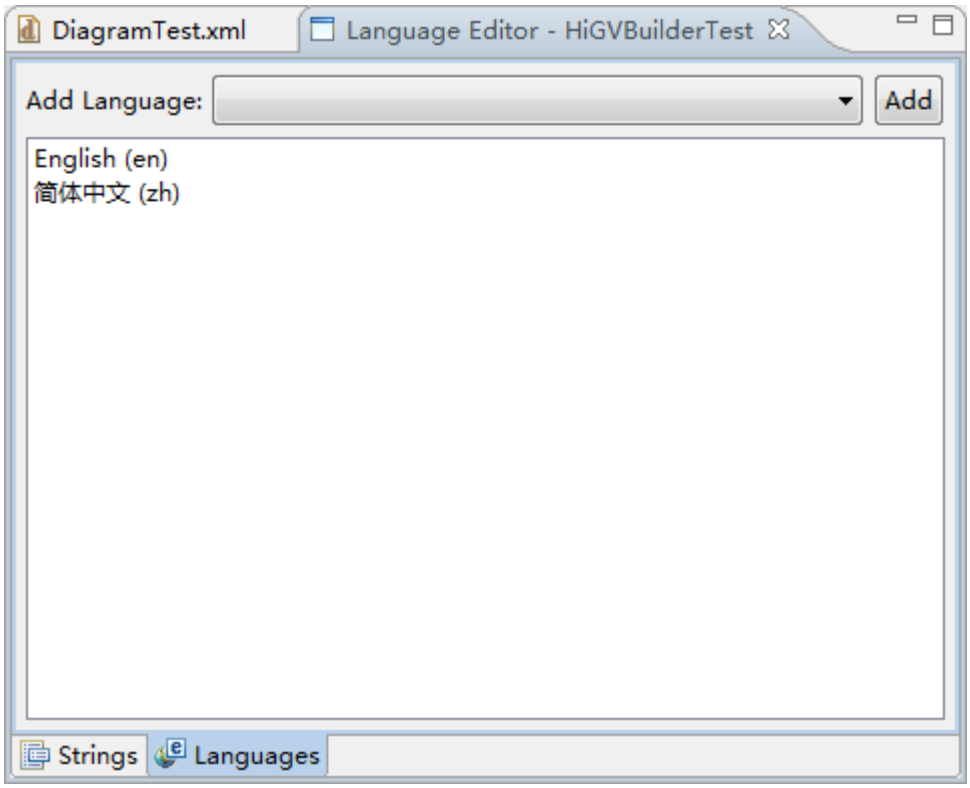
CAUTION

You cannot delete the multi-language character strings that are being used.

3.2.5.3 Managing Languages

To manage various languages, click the **Languages** tab at the bottom of the language editor.

Figure 3-16 Language management GUI



Then you can manage various languages.

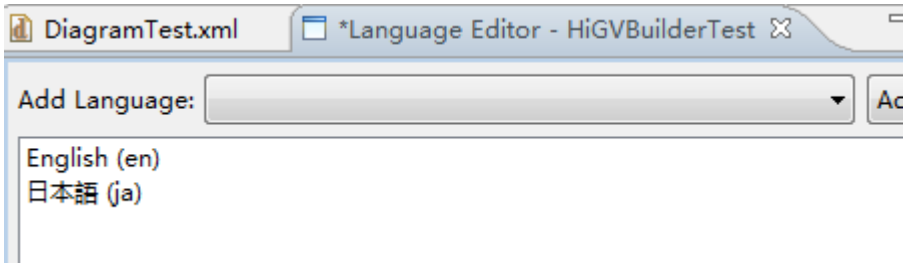
The **Add Language** drop-down list contains all languages that can be added (including all languages listed in Table 3-1. The languages that have been added to the list box below are not displayed in the drop-down list.). To add a language, select a language, and click **Add**.

Figure 3-17 Adding languages

To delete a language, right-click the language in the list box, and choose **Delete Selected Language(s)** from the shortcut menu.

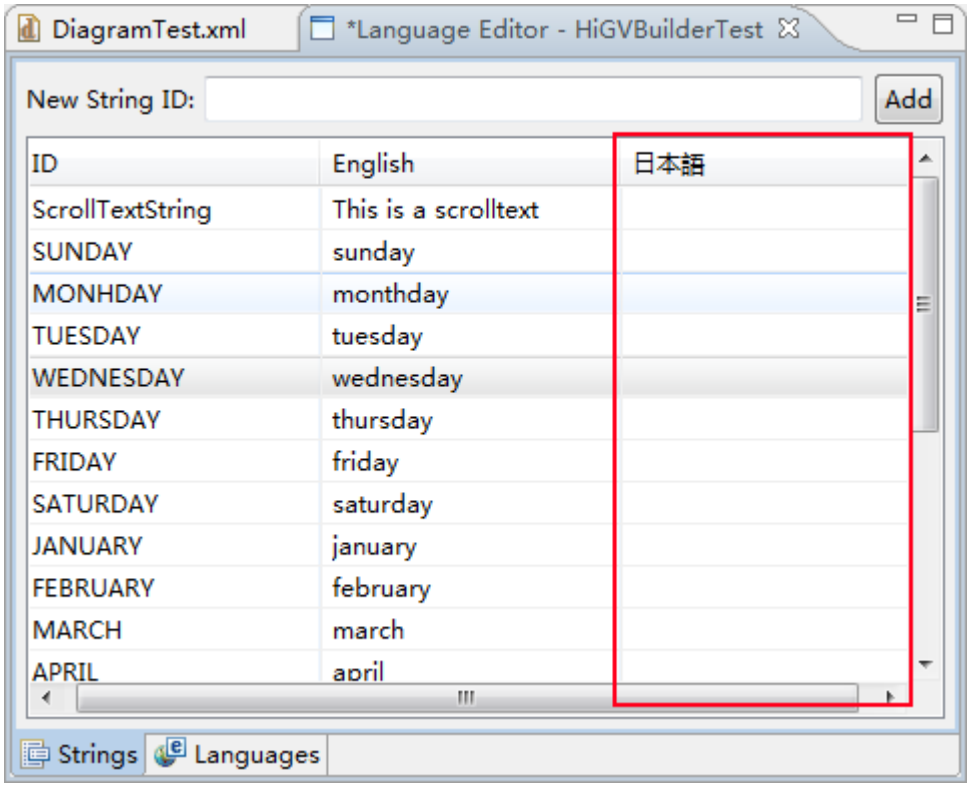
For example, if you add the Japanese and delete the Chinese, the list box is displayed shown in Figure 3-18.

Figure 3-18 Adding and deleting a language 1



When you switch to the **Strings** tab page, you will find that the **Japanese** column is displayed instead of the **Chinese** column.

Figure 3-19 Adding and deleting a language 2



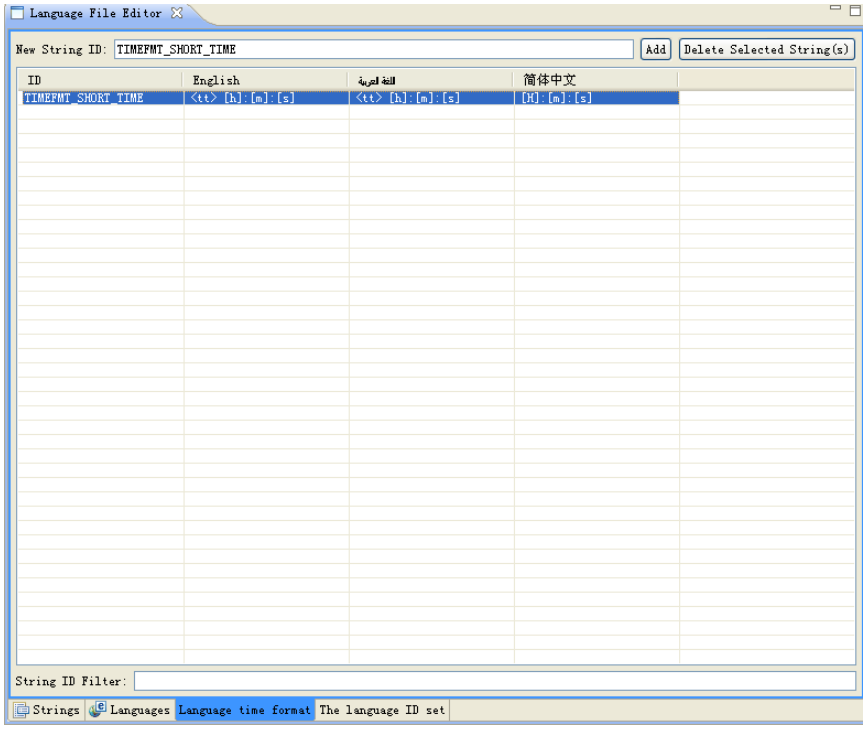
CAUTION

You cannot delete a language if it is the only one left in the list box.

3.2.5.4 Setting the Multi-Language Time Format

To manage the multi-language time format, click the **Language time format** tab at the bottom of the language editor, as shown in Figure 3-20. Enter a new string ID, and press **Enter** or click **Add** to add the ID to the list. Then you can directly edit the multi-language time format.

Figure 3-20 Setting the multi-language time format



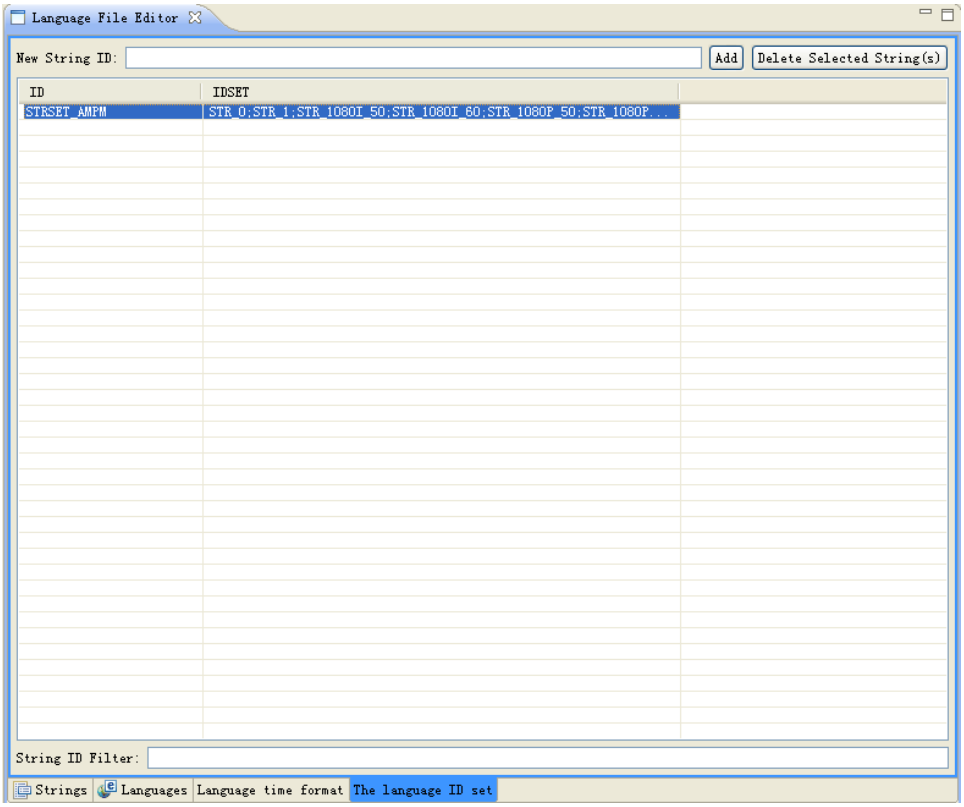
NOTE

The time format must comply with the standard of the clock control.

3.2.5.5 Setting the Multi-Language Character String Set

To set the multi-language character string set, click the **The language ID set** tab at the bottom of the language editor, as shown in Figure 3-21. Enter a new string ID, and press **Enter** or click **Add** to add the ID to the list. Then you can directly set the multi-language character string set.

Figure 3-21 Setting the multi-language character string set



NOTE

Contents in the **IDSET** column such as **STR_0** and **STR_1** are definitions of the multi-language character strings.

3.2.5.6 Importing Language Configurations from a File

To import languages for a project, right-click the **Languages** directory of the corresponding project in the Resource Manager, choose **Import from File** from the shortcut menu, and select the file to be imported in the displayed dialog box. The languages can be imported from an XML or Excel file.

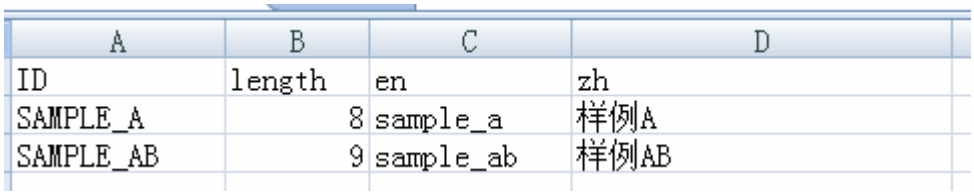
If the imported language file is an Excel file, it must comply with the basic format standard.

- The first row specifies the auxiliary functions and do not need to be filled.
- The second row specifies the column headings:
  - The first column is the remarks information and do not need to be filled.
  - The second the third columns must be **ID** and **length** (case-sensitive).

    length specifies the maximum length of the language character string.
  - The fourth column is the remarks information and do not need to be filled.
  - The fifth and seventh columns are identifiers of specific languages (for example, zh and en, as listed in Table 3-1).
  - The sixth column can be filled with only Arabic characters (the column does not need to be filled if there are no Arabic characters).
- Rows starting from the third are specified based on the headings in the second row.

Figure 3-22 shows an example.

Figure 3-22 Importing language configurations from a file



A	B	C	D
ID	length	en	zh
SAMPLE_A	8	sample_a	样例A
SAMPLE_AB	9	sample_ab	样例AB

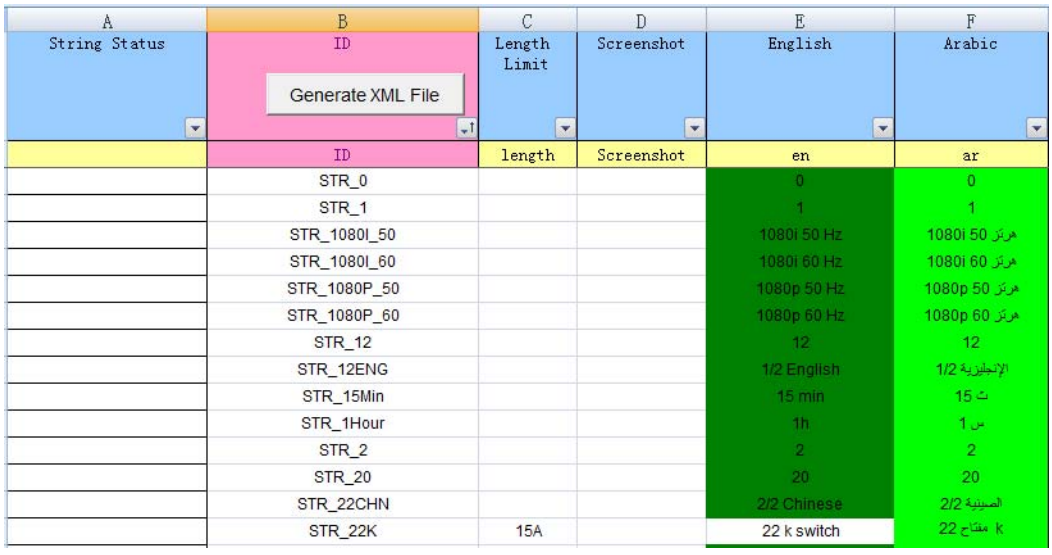
CAUTION

- Currently the HiGVBuilder supports the import of only Excel 97-2003 files.
- The Excel file template **Language-template.xls** is stored in the **source\public** directory.

3.2.5.7 Exporting Existing Files from Language Configurations

To export language files for a project, right-click the **Languages** directory of the corresponding project in the Resource Manager, choose **ExportExcel** from the shortcut menu, and select the file to be exported in the displayed dialog box.

Figure 3-23 Exporting existing files from language configurations



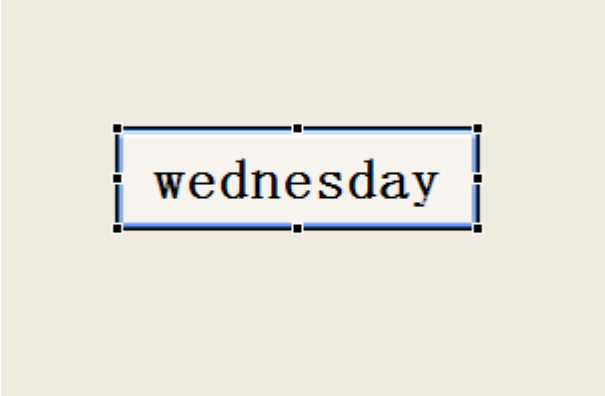
A	B	C	D	E	F
String Status	ID Generate XML File	Length Limit	Screenshot	English	Arabic
	ID	length	Screenshot	en	ar
	STR_0			0	0
	STR_1			1	1
	STR_1080I_50			1080i 50 Hz	1080i 50 هرتز
	STR_1080I_60			1080i 60 Hz	1080i 60 هرتز
	STR_1080P_50			1080p 50 Hz	1080p 50 هرتز
	STR_1080P_60			1080p 60 Hz	1080p 60 هرتز
	STR_12			12	12
	STR_12ENG			1/2 English	1/2 الإنجليزية
	STR_15Min			15 min	15 ث
	STR_1Hour			1h	1 س
	STR_2			2	2
	STR_20			20	20
	STR_22CHN			2/2 Chinese	2/2 الصينية
	STR_22K	15A		22 k switch	22 مفتاح k

3.2.5.8 Setting the Display Language of the Current Project

In the HiGVBuilder, you can select a language to enable all controls that display texts in the GUIs being edited to display character strings in this language. This operation can be implemented in the Resource Manager.

To set a language as the display language of the drawing being edited, right-click the language and choose **Set as Display Language** from the shortcut menu. Figure 3-24 shows the display effect when English is selected as the display language.

Figure 3-24 Display effect when English is selected as the display language



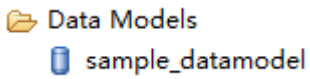
3.2.6 Data Model

3.2.6.1 Introduction

The data model is used to specify the data source of data type controls (such as the ComboBox). The HiGV reads data required by the user and fills the data into the controls based on the sources and configured parameters in the data models.

The **Data Models** directory of each project in the Resource Manager lists the defined data models in the corresponding project, as shown in Figure 3-25.

Figure 3-25 Data model list



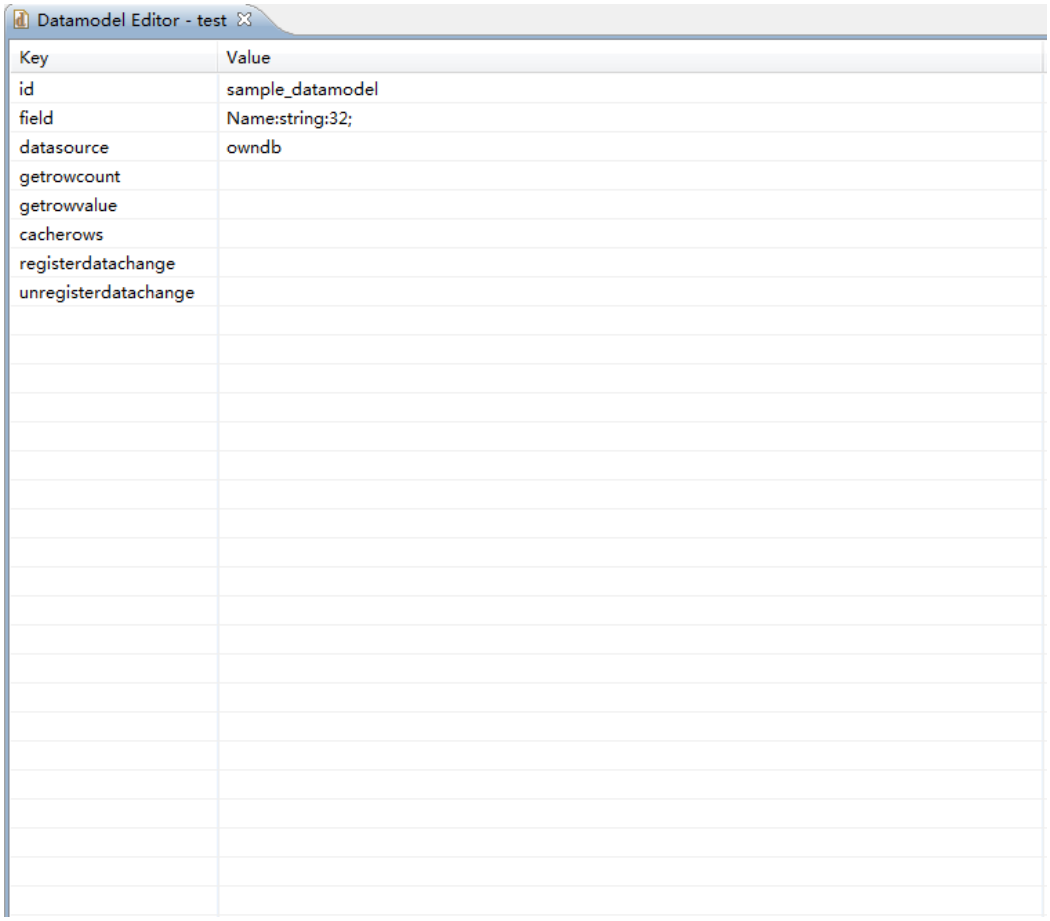
3.2.6.2 Creating a Data Model

Right-click the **Data Models** directory of a project, and choose **Create Data Model** from the shortcut menu. The dialog box for creating a data model is displayed.

- **ID**: Specifies the unique ID of the data model.
- **Field**: Describes field information.
- **DataSource**: Specifies the data source type (**owndb** indicates the internal DB, and **userdb** indicates the user DB).
- **Cacherows**: Specifies the number of cache rows.
- **GetrowCount**: Obtains the total row count.
- **GetrowValue**: Obtains row data.
- **Registerdatachange**: Registers the data model handle with the dependent user DB.
- **Unregisterdatachange**: Deregisters the data model handle with the dependent user DB.

After filling all the information, click the save button on the toolbar to add the new data model to the data model resource library of the project. You can view the newly added data model in the main GUI.

Figure 3-26 Adding a data model



3.2.6.3 Modifying a Data Model

To edit a data model, double-click the icon of the data model. The operations for modifying a data model are the same as those for adding a data model.

3.2.6.4 Deleting a Data Model

To delete a date model, right-click the data model and choose **Delete**. You can delete multiple data models at the same time.

CAUTION

Data models that are being used cannot be deleted.

3.2.6.5 Importing a Data Model

To import a data model, right-click the **Data Models** directory of a project, choose **Import Data Model** from the shortcut menu, and select the data model XML file in the displayed dialog box.

3.3 Editing the GUI

3.3.1 Introduction

To edit a GUI, you need to select the controls to be generated in the tool area, drag and generate the controls in the editing area, and edit the controls in the editing area and property area. Then you edit and combine all the controls to generate the required GUI.

3.3.2 Editing the GUI

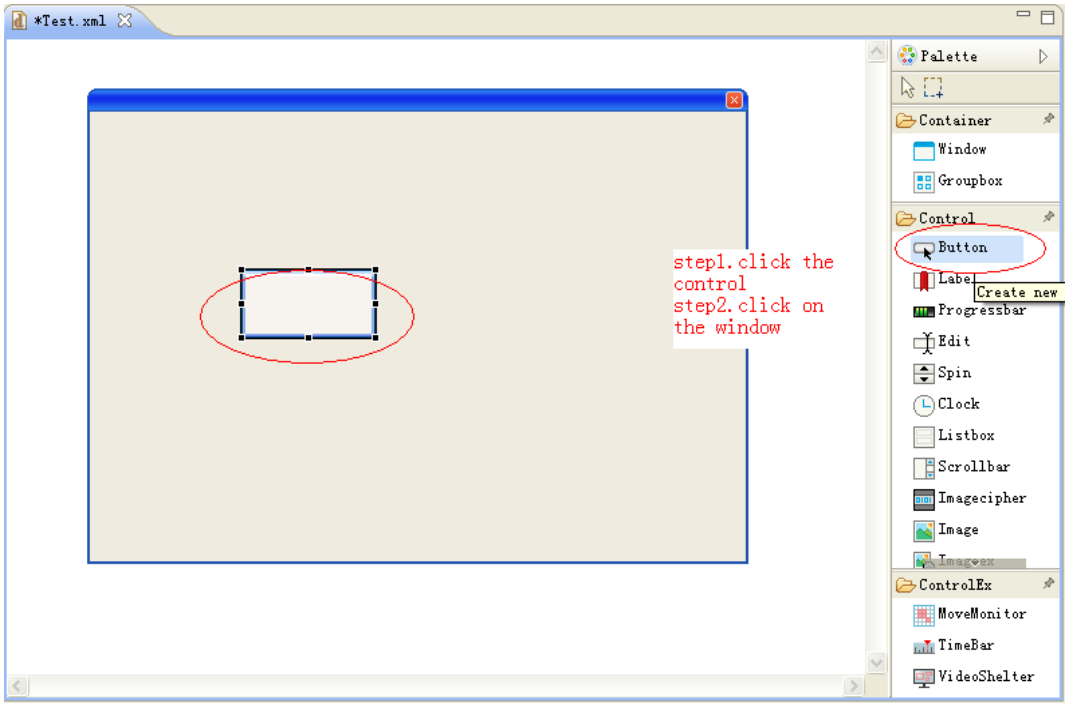
3.3.2.1 Opening the Editing Area

When you create a GUI XML file, the HiGVBuilder opens the editing area by default, and the tool area is opened or closed with the editor. You can also double-click the corresponding GUI XML file in the Project Explorer to open the editing area.

3.3.2.2 Generating Controls

After selecting the control to be generated in the tool area, you can click the editing area to generate a control of the default size, or you can drag the control in the editing area to generate a control of the required size (identified by a yellow block).

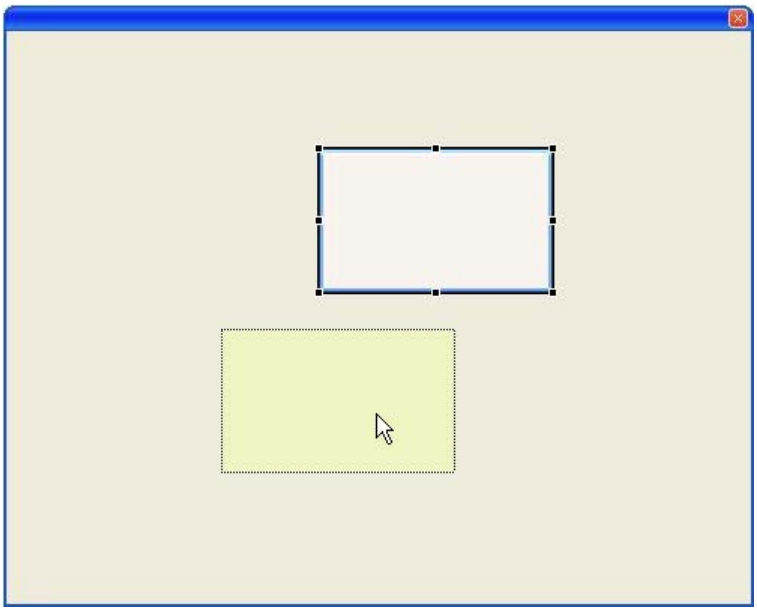
Figure 3-27 Generating a control



3.3.2.3 Dragging Controls

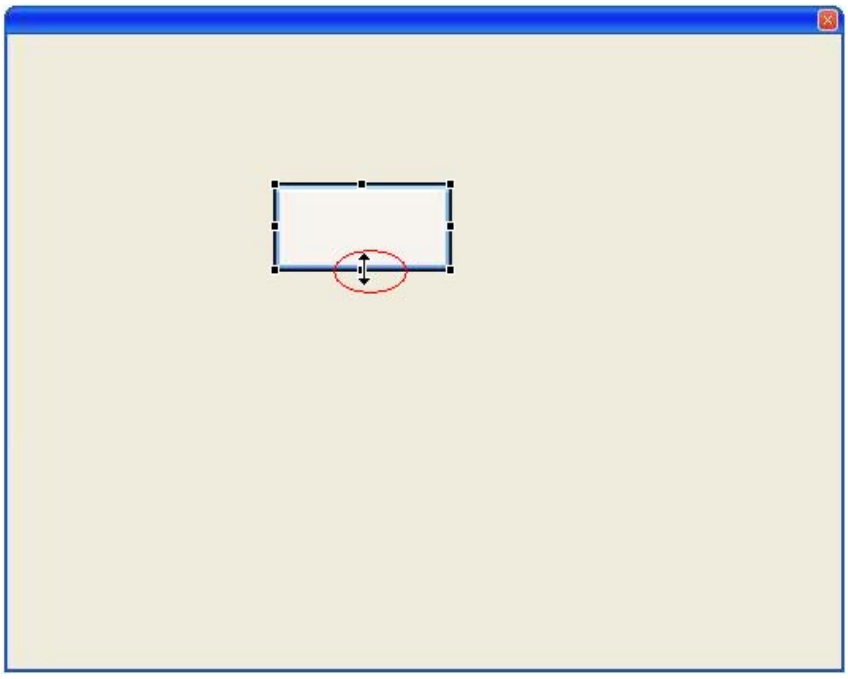
The drag, scale, and edit operations on controls in the editing area are implemented by using the mouse (or shortcut keys such as the direction keys). To drag a control, select the control, move the mouse to drag the control to the destination position (for example, the yellow block in Figure 3-28), and then release the mouse. The control is moved to the new position.

Figure 3-28 Dragging a control



To change the control size, select the control first. The handle identifier is displayed around the control. When the mouse is moved over the eight black points around the handle, the mouse icon turns into a bidirectional arrow. Then you can move the mouse to drag the control to the required size and release the mouse. You can also drag any of the four black points in the four corners to change the control width and height at the same time.

Figure 3-29 Changing the control size



A control can be dragged from one container to another container. "—>" indicates the attaching relationship.

In the same window:

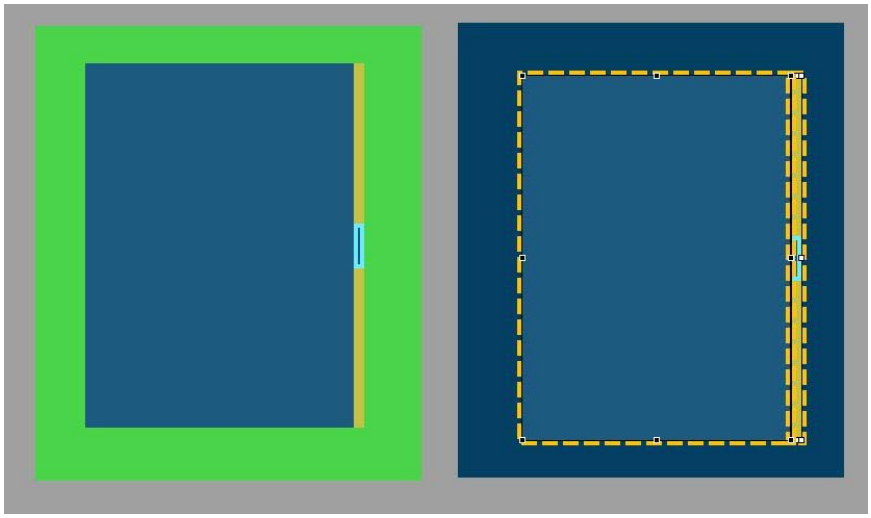
- Group——>window
- Group1——>group2

In different windows:

- Window1——>window2
- Window1——>window2.group
- Window1.group——>window2
- Window1.group——>window2.group

After a control is bound to a ScrollBar, if you want to change the parent container of the control, you need to select both the control and the bound ScrollBar and then drag them to the destination parent container. See Figure 3-30.

Figure 3-30 Dragging a control with a ScrollBar to a new parent container



TIP

These auxiliary functions are provided through the toolbar buttons. Some shortcut keys are provided, such as **Ctrl+Z** (cancel) and **Ctrl+Y** (redo).

3.3.2.4 Hiding/Displaying Controls

In the actual application, some complex GUIs may contain many child control windows and group boxes, and controls of various containers may overlay. Therefore, editing one control or container among many other controls seems inconvenient.

To solve this problem, you can hide the controls that do not need to be edited by using the hide/show function of the Outline view. For details, see section 2.5.3 "Outline View."

3.3.2.5 Editing Properties

Most of the edit operations on controls are performed in the property area. The properties can be divided into the following types:

- Properties with several fixed values: This kind of properties mainly includes the type and Boolean value properties, for example, style (normal, check, radio). You can select the values from a drop-down list.
- Properties with many fixed values: This kind of properties mainly includes the resource properties, such as the skin. You can select the required resource from the displayed list box and then click **OK** or press **Enter**, or you can double-click the required resource.
- Properties that allow the user to select a local file or resource, such as a picture. You can select the resource by using the file selector in the displayed window.
- Properties that are automatically generated, such as the event property and ID property of controls. The properties can also be modified.
- Properties whose values are manually entered, such as the position and size of a control. The input values are automatically verified. If the input values are invalid, an instruction message is displayed. The characters in red in the status bar are the instruction information for user inputs.
- Properties whose values are within a specific range but the number of values are not fixed, for example, property of the brotherobj or scrollbar. You can select values from the drop-down list, but the content in the drop-down list is not fixed.
- Other properties, such as the content alignment property for the button control, item property for the ListBox or ScrollBar, and highlight color property for the calendar control (calendar-satcolor). For the color button, you can select colors from the system palette or manually enter hexadecimal color values.

TIP

The properties with no identifier (identifier for the displayed property dialog box and drop-down list) need to be set by entering values.

For details about the control properties, see section 3.3.3 "GUI Elements and Controls."

3.3.2.6 Auxiliary Functions

The HiGVBuilder provides the following auxiliary function for flexible GUI design:

- Undo: Cancels the previous operation.
- Redo: Restores the operation that is canceled.
- Align Left: Aligns all selected controls with the left edge of the last selected control.
- Align Right: Aligns all selected controls with the right edge of the last selected control.
- Align Vertical Center: Aligns the vertical central lines of all selected controls with that of the last selected control.
- Align Top: Aligns all selected controls with the top edge of the last selected control.
- Align Bottom: Aligns all selected controls with the bottom edge of the last selected control.
- Align Horizontal Center: Aligns the horizontal central lines of all selected controls with that of the last selected control.
- Align Width: Equalizes the horizontal spacing between selected adjacent controls (three or more).
- Align Height: Equalizes the vertical spacing between selected adjacent controls (three or more).
- Match Width: Equalizes the width of all selected controls based on the width of the last selected control.
- Match Height: Equalizes the height of all selected controls based on the height of the last selected control.

- Format Painter: Enables/Disables the format painter (**Alt+B**). When you click the format painter button, the skin and font configurations of the selected control are copied to the clipboard. Then you can click another control in the GUI to overwrite the skin and font configuration of the control with the formatting in the clipboard. You can select the formatting to be overwritten from the drop-down list in the right. Note that the currently supported skins are the five skins shared by all controls.

  The following describes the formatting overwriting policies corresponding to the format painter states (the three states can be switched by clicking the format painter button on the toolbar).

   : The format painter is disabled, and no formatting will be overwritten.

   : The formatting of the selected control is applied to the control that is selected next. Then the format painter is automatically disabled.

   : The formatting of the selected control is applied to all the controls that are selected afterwards until you disable the format painter manually.
- Scale: Zooms in/out on the entire editing GUI proportionally.

TIP

These auxiliary functions are provided through the toolbar buttons. Some shortcut keys are provided, such as **Ctrl+Z** (Undo) and **Ctrl+Y** (Redo).

NOTE

The alignment and width, height, and spacing equalization operations can be performed only when multiple controls of the same level (in the same parent container) are selected.

3.3.2.7 Opening the XML Editor

After editing the required GUI, you need to convert the GUI into an XML file using the HiGVBuilder by saving the edited controls.

To view and edit an XML file, right-click the XML file in the Project Explorer, choose **Open with** > **Text Editor**.

3.3.3 GUI Elements and Controls

The GUI element is the basic unit for describing the GUI by using the descriptor, property, and event. Each GUI element has its unique descriptor. For details about the definitions of XML file label attributes in the HiGV, see the *HiGV Label User Guide*.

Table 3-2 GUI elements in the HiGVBuilder

Name	Descriptor	Remarks
View	view	-
Window	window	-
Skin	skin	-
Font	font	-
GroupBox	groupbox	-
Button	button	-

Name	Descriptor	Remarks
Label	label	The background, text, or background plus text are supported.
ImageBox	image	-
ImageEx	imageex	The dynamic .gif images are supported.
Data model	datamodel	
ListBox	listbox	-
ScrollBar	scrollbar	-
EditBox	edit	-
Spin	spin	-
ProgressBar	progressbar	-
ScrollText	scrolltext	-
ImageCipher	imagecipher	-
Clock	clock	-
ScrollGrid	scrollgrid	-
Language	language	-
ScrollBox	scrollbox	-
ComboBox	combobox	-
Calendar	calendar	-
TrackBar	trackbar	-
IPEdit	ipedit	-
TimeTrack	timetrack	-

3.4 Saving and Parsing XML Files

3.4.1 Overview

The HiGVBuilder can save the GUI as an XML file, which contains GUI information such as controls, container relationship, coordinates, and sizes. It can also contra-parse the XML file to allow you to continue to edit the GUI.

3.4.2 Saving a GUI as an XML File

After editing the GUI, you can click the save button () on the toolbar or press **Ctrl+S** to save the GUI information in an XML file. The HiGVBuilder also provides the

function of automatically saving the GUI. You can choose **File** > **AutoSave**, and then set the auto-save interval.

3.4.3 Parsing XML Files

Right-click a project in the Project Explorer, and choose **Import Ui Layouts** from the shortcut menu. The dialog box for importing the GUI XML file is displayed. Click **Browse**, select the XML file to be imported, and click **Finish**. Then the HiGVBuilder automatically opens the corresponding GUI.

If you need to import multiple XML files at a time for parsing, copy the XML files to be parsed to the XML directory of the current project. Then you can double-click an XML file to open the corresponding GUI.

4 Precautions

4.1 Importing Files to the HiGVBuilder

Before importing graphics files and common resources (fonts, data models, language character string tables, and skins), ensure that the files to be imported meet the following requirements of the HiGVBuilder:

- If the imported file is an XML file, the encoding format must be UTF-8. Otherwise, the file cannot be edited, or cannot be identified by the XML2BIN tool.
- If the imported file is an XML file, it must comply with the W3C standard version 1.0 (for details, see http://www.w3.org/TR/REC-xml/). Multiple top-level elements are supported because common resources must support the HiGV specifications (other aspects must comply with the standards defined in W3C).
- When a language file is imported, the language identifier (consisting of two letters) must match the language identifier defined by the HiGV. Otherwise, it cannot be correctly identified. For details about the language identifiers, see Table 3-1.
- If the imported language file is an Excel file, it must contain the **ID** and **length** columns as the first and second columns.

4.2 Managing Projects

You are advised not to directly copy files to the directory for storing projects or modify files in a project directory externally. Otherwise, problems may occur due to the buffer mechanism and fault tolerance mechanism of the HiGVBuilder. If you want to import resources or drawing from outside the project directory, use the import function of the HiGVBuilder.

5 FAQs

- Q: What do I do if the control skin cannot be displayed when a GUI file is imported?

  A: When the HiGVBuilder imports a GUI file, it does not import the skin used by the GUI at the same time. You need to import the skin file as well. For details, see section 3.2.3.5 "Importing a Skin." In addition, if the skin type is image, ensure that the path for the used image is correct. If the image path is incorrect, modify it by using the skin editor.

- Q: What do I do if an error message similar to "Element : Window -> Attribute : alignment is unknown Invalid argument." is displayed when the imported GUI file is opened?

  A: If this kind of error message is displayed, some property values in the imported XML file are not supported by the corresponding controls. For example, if "Element : Window -> Attribute : alignment is unknown Invalid argument." is displayed, the window container does not support the configured alignment property. In this case, you can open the XML file by using the Notepad, delete the incorrect property values, and save the file. Then the GUI file can be opened in the HiGVBuilder.

- Q: What do I do if an error message similar to "Invalid diagram XML file. Invalid byte 1 of 1-byte UTF-8 sequence." is displayed when the imported XML file is opened in the HiGVBuilder?

  A: This kind of error message is displayed because the XML parser of the HiGVBuilder cannot properly identify the file. The HiGVBuilder requires that all XML file be saved in the UTF-8 encoding format. If this kind of error occurs, open the XML file in a text editing tool such as the Notepad, and save the file in the UTF-8 encoding format. Then you can open the XML file properly in the HiGVBuilder.