# The easyGUI software explained

## The easyGUI packages

easyGUI is delivered in three different packages, with varying capabilities:

easyGUI Monochrome supports monochrome displays (1 bit per pixel), i.e. displays where each pixel is either on or off. Grayscale displays (4 or more gray levels) are considered color displays by easyGUI.

easyGUI Color includes all functionality of easyGUI Monochrome, and adds support for grayscale and color displays, from 4 gray levels / colors (2 bits per pixel) up to 16 million colors (24 bits per pixel, like Windows true color). easyGUI Color supports both monochrome, grayscale, and color displays.

easyGUI Unicode includes all functionality of easyGUI Color, and adds support for Unicode characters (16 bit character codes), opening up for the use of virtually all languages in the world, including Asian and Middle East languages. easyGUI Unicode can also be used for projects using standard ANSI characters (8 bit character codes). easyGUI Unicode supports both monochrome, grayscale, and color displays.

The basic functionality in easyGUI - that is font management, structure creating and editing, translation, etc. is the same across all easyGUI packages.

# Major features of easyGUI packages

## Projects

One easyGUI project can be open at any time. Each easyGUI project contains all data necessary for the visual user interface of an embedded project.

All data for an easyGUI project is contained in a single database file, for easy backup. This file uses the same format for all easyGUI variants - it is thus easy to later upgrade from easyGUI Monochrome to one of the more advanced packages. Internally a highly stable and efficient relational SQL database is used.

## Fonts

easyGUI use bitmapped fonts, for fast and efficient display writing. All fonts by definition use proportional writing (varying character widths like in e.g. Windows). Fixed spacing character writing can also be used, if desired.

An unlimited number of fonts can be created in a project. What fonts are actually included in the target system is determined on a font by font, and character by character basis. It is thus possible to make efficient use of memory resources in the target system. Furthermore, font data are stored in a highly compressed format, to ensure an effective combination of lowest possible memory usage and fastest possible text rendering.

A diverse number of fonts are supplied with the easyGUI packages, in various sizes and styles.

## Font editor

The integrated easyGUI font editor allows complete handling of fonts. New fonts can be created from scratch, or converted from Windows TrueType/OpenType fonts, or the supplied fonts can be used as is.

Each font can contain 256 characters (ANSI character mode) or 65536 characters (Unicode mode).

## Display setup

Any practical display size can be specified. The display can freely be rotated between the four primary display orientations: Normal, 90° left, 90° right, and upside down. This allows alternative mounting of displays without any problems. Similarly, displays can be mirrored horizontally and vertically, making it easy to correct situations where the scan line direction is the opposite of expected.

## Compiler setup

Several customization options are possible for compiler setup, in order to handle special compilers which are not ANSI X3.159-1989 Standard C compliant. This is unfortunately the case for quite a number of commonly used embedded world compilers, and is thus handled with ease.

## System setup

Parts of the embedded easyGUI library can be turned off, if memory is very constrained. This of course limits the functionality available in the target system, but never the less allows advanced graphical interfaces to work on small memory limited systems.

## Structures

The core entity of the target system user interface is the structure. A structure is a collection of items, where each item is a simple object, like a text, a variable, a geometrical figure,  a bitmap, etc.

A complete screen on the target system can be contained in a single structure, or divided into separate structures for e.g. headline, menu commands, and main functionality.

This is entirely up to the developer to decide.

A structure can also be a sub-assembly of items, which can be called from many other structures. Often used components of the user interface, like e.g. an error box, or an edit field, can be created as a single instance, and then used many times throughout the user interface, as desired.

Structures can call other structures in any number of levels, either directly, or controlled through a variable.

The latter allows dynamic behavior to be built right into the structures - this is one of the strong features of the easyGUI system.

A comprehensive number of features allow total control over structures - their creation, construction, and interaction. Everything is controlled in the Structures editor window, and this is a central part of the easyGUI PC application.

## Languages

easyGUI fully supports multilingual target applications. When more than one language is necessary in the user interface easyGUI offers comprehensive features to handle this.

A translation window assists in translation between languages, with visual representation of display result in both original and translated languages. Any number of languages can be added to an easyGUI project. For each language it can be specified if the preferred writing direction is left-to-right or right-to left.

Translation tasks can also be performed externally to easyGUI, see the easyTRANS utility section below.

## Display positions

A list of fixed position coordinates across the complete GUI can be maintained in the easyGUI project file. These coordinates can be used to ensure that related items are placed at the same position in different structures, e.g. the vertical position of a headline. A central adjustment can then be made, potentially modifying a large number of structures at once.

## Variables

C variables can be created in the easyGUI system, and used for displaying variable values, or for controlling dynamic aspects of the user interface, through indexed calling of other structures from a structure.

## C code generation

The final stage in easyGUI is generation of the actual target system C code. A number of files are generated, containing general setup, font data, structure data, bitmaps, and variables created from within easyGUI. The generated c and h files are part of the target system code, along the easyGUI library files, the easyGUI display driver, and the usual developer supplied files.

## Import / export

To facility exchange between easyGUI project an Import/export feature allows selected types of data (fonts, structures, setups, etc.) to be exchanged.

Graphical user interfaces has never been easier, to learn more see our website www.easygui.com, or contact us at sales@ibissolutions.com

# The easySIM feature

easySIM is a PC simulator which is part of the easyGUI package. It is, in essence, a special display driver working in the Windows 32 bit API environment, instead of driving an embedded display in the target system. easySIM must be combined with a suitable C++ compiler generating PC applications, like Microsoft Visual Studio, Borland C++ Builder, or DEV C++ (freeware under the GNU license). There are many ways of utilizing easySIM:

- Testing GUI designs at an early stage. Testing in the correct resolution and color depth is vital for judging whether GUI design elements will work as intended.
- Creating simulator executables for marketing, sale, service, manual writing, etc.
- Developing the full GUI before actual hardware is at hand. This can substantially shorten development times.

## The easyTRANS utility

easyGUI supports multilingual target applications, and handles translation between languages very well. However, because the Translator is typically not a developer it can be more convenient to have a separate application for the translation job, this is the easyTRANS utility. There are two advantages to this arrangement:

1. The Translator can work with easyTRANS anywhere in the world, without needing an easyGUI license.
2. The Translator can only translate text. The translator will not be able to edit anything else in easyGUI.

easyTRANS does not contain any functionality not found in easyGUI, but it allows for remote translation to take place. In cases where a professional translation agency or consultant is needed, this tool comes in very handy.

The procedure for using easyTRANS is straight forward:

1. An export file is made in easyGUI. The exported file, a translate file, contains the complete easyGUI project in two languages, selected in easyGUI, a reference language and the language to be translated to.
2. The translate file is handed over to the Translator, along with the easyTRANS install program, if the Translator does not already have easyTRANS installed.
3. The Translator opens easyTRANS, and selects the relevant translate file.
4. The Translator goes through all texts, and translates them. For each text a graphic representation of the screen(s) using this particular text is shown, both in reference language and in translated form. This enables the Translator to see the resulting translated text in its context, and resolve eventual problems with space constraints.
5. The translate file is handed back to the Developer.
6. The Developer imports the translate file back into the easyGUI project, and checks for possible inconsistencies and misunderstandings.

easyTRANS is part of the easyGUI package.