Introduction to U++

Welcome to U++.

U++ is both an application framework (class library for Win32 and Linux) as well as a complete development tool including RAD GUI designer, source editor, image editor, code documenting tool, and C++ project builder. It also provides a powerful "C-like" scripting language (ESC) enabling you to provide end user extensibility and customization of your own applications, as well as extend TheIDE (the U++ IDE)

This topic describes the key concepts of U++ to help you get started as quickly as possible. If you want to try out U++ before reading any further, you can build and run the HelloWorld example and read the HelloWorld tutorial here (fix).

TheIDE

Before you can run TheIDE, you must first choose the package and assembly you wish to work with using the "Select-main-package" dialog. After an assembly and package have been selected, the package is opened in TheIDE. The package that has been opened is referred to as the **main package** and its name is shown in the title bar of the U++ application window. The uppermost pane at the left hand side of TheIDE shows the name of the main package followed by a list of the packages used by the main package. The lower left-hand pane lists the source files belonging to the package whose name is highlighted in the upper pane. To select a different main package, the "Set main package" option from the File menu is used. For more detail on using TheIDE, see this.

Packages, assemblies and nests

Packages are centric to U++. An executable application is built from a package. A package can also build into a dynamic link library, a static library, or a set of object files. A package can be used by other packages. A package corresponds to a single directory whose name is the name of the package. TheIDE itself is a package and you can rebuild it from within TheIDE if you wish!

An assembly can be thought of as a collection of packages but it is actually just a set of paths which determine where U++ looks for the packages and source files needed when building one of the assembly packages. The paths defined by the assembly are stored in an assembly definition file which has a .var extension and is stored in the U++ root installation directory. A package can "appear" in multiple assemblies.

U++ requires that packages be organized into nests. A nest is actually just a directory containing a set of package directories, folders or files. An assembly defines an ordered list of nests (paths) and the packages contained in those nests form the packages of the assembly. For more detail on packages and assemblies see "Packages, Assemblies and Nests" and. "Creating and Configuring Assemblies and Packages"

RAD GUI designer

The ESC macro language

The help system.

U++ documentation is created with Topic++.

U++ License

The U++ license is BSD, allowing almost unrestricted use of the product and its source code. Full C++ source code is provided for both theIDE and the class framework, ensuring both the future of U++ and the ability to support and extend your own applications.