Author: J.F.Benckhuijsen (jfbenck@users.sourceforge.net)
Version: 1.0

MDK Project Library

1. Introduction

The whole MDK application was designed to be easily portable to other platforms. Because of this, large parts of the library were written in standard C++. Besides that, we moved as much code from the GUI to other files.

The GUI is the non-portable part of the application. So when the application will be ported, the UI is the only thing that has to be implemented again. To speed up this implementation process, we've created the MDK Project library. This library acts as a façade for the application libraries. This class handles loading, saving, storing of options, etc., and provides a easy-to-use entry point for the UI.

2. Provided functionality

The MDK Project library consists out of two classes. The first is the TMdkProject class, the functional part of the library. The oher class: TTestSystem is used for debugging purposes. As the name states, it creates a Testing environment. It creates a new project and some data to test the application with.

The TMdkProject class is off course the most interesting class in the library. This class offers a lot of high-level functionality to the application:

- Methods to build the different models MDK currently supports;
- High level application functions like loading and saving;
- Storing of project level options, like project name and filename;
- Storing of generation options, like UI colors, output directory;
- Managing the Kiss System and Kiss Rules.

Besides that it allows a User Interface equivalent to be created which can also store it's User Interface specific data, using two virtual methods.