

**Author:** J.F. Benckhuijsen ( [jfbenck@users.sourceforge.net](mailto:jfbenck@users.sourceforge.net))

**Version:** 1.0

## File IO Design

### 1. Introduction

In most applications, you'll want to save your work. Of course, the MDK Application is also one of those. The purpose of saving your work is you can restore it later. You'll want to save the complete status of the program, so you'll be able to continue where you stopped.

### 2. File format

XML is rapidly becoming a world-wide standard to store all different kinds of data. Because of the availability of many XML processors and the portability of the file format itself and its processors, we've chosen to store our data in XML.

When saving a project we'll have to store three different kinds of data, for which a branch is created in the XML tree.

1. The system itself (<System> branch)
2. The generator and project options (<Options> branch)
3. The user interface data (<Interface> branch)

So the file structure will be:

```
<?xml version="1.0">
<MDKProject>
    <System>
        // System related data goes here
    </System>
    <Options>
        // Options data goes here
    </Options>
    <Interface>
        // User Interface data goes here
    </Interface>
</MDKProject>
```

### 3. Implementation

Currently the whole File IO implementation is written using the gnome-xml library. This is a portable XML processor. The disadvantage of using this processor is, that the windows binary applications size is increased. In Linux there is a shared equivalent of this library, so the binary will remain small. To allow the same to work in windows we could create a dll version of gnome-xml, or we'll be using the Microsoft XML parser included with windows. The latter case saves us some installation problems, we'll only have to copy the executable. However we'll have to write a generic frontend for both Microsoft XML and gnome-xml to retain the same level of portability.