**AIISRR** Artificial Intelligence Infinity Software Reliability Regulation

AIISRR-C-: 2010

Regulations for the use of the C language in critical system

1. 开发工具
   1. 代码编辑器

就目前而言AII建议采用较新版本的Microsoft Visual Studio和其环境增强插件WTS Visual Assist X作为代码编辑器并且采用Source Insight作为代码浏览工具。其余编辑器如VIM, EMACS, eclipse以及Ultra Editor也可以使用，但是必须配置成与VS+VA兼容的设置。出于开发效率和项目兼容性考虑，Microsoft Visual Studio 2008 + WTS VAX是目前AII大部分项目的标准开发工具。

* 1. 编译器

**Windows Platform:** 唯有MS Visual Studio Compiler 和 Intel Compiler可保证一直用于一般性项目开发。任何一个MS Visual Studio Compiler和Intel Compiler版本必须经过AII技术委员会定期认证方可用于特定项目开发。可在Windows平台运行的其余编译器必须经过AII技术委员会认证方可用于项目开发。

**Linux Desktop Platform:** 唯有GCC可被AII技术委员会定期认证并可用于一般性项目开发。一些基于Intel平台的项目可采用Intel编译器Linux版本，但必须经过AII技术委员会认证方可用于项目开发。

**Linux Embedded Platform:** 任何项目不得使用开源Linux编译器，商业版编译器必须经过AII技术委员会认证方可用于项目开发。当前而言Windriver System Linux解决方案以及QNX解决方案可用于AII的嵌入式项目开发。

**vxWorks Embedded Platform:** 唯有vxWorks6.0以上的版本才能用于AII嵌入式项目开发。

**MAC OS X Platform:** 唯有xcode开发平台才能用于AII的项目开发。

* 1. 项目版本管理

AII recommends Subversion for small scale projects and git for large scale projects. AII will introduce commercial version management solution later.

1. Project Structure
   1. General Folder Structure

The project folder should contain at least three directories: build, src, doc, whatever it is used for any type of project and in any development phase.

The folder “build” contains project definition files which are solution setting files in Visual Studio of Windows platform, Makefile in GCC platform or other project setting files of other platform. Different platforms must settle on different folders and name them with platform name. Standard name must be “linux”, “win32”, “macos”, “vxWorks”.

The folder “src” contains all the C source code and header files. The internal structure of “src” folder are slight different in different project type.

The folder “doc” contains all the development documents of the project. The development documents should be cataloged into different folder. Only MS office format and PDF are acceptable .

* 1. Application Integration Development

The folder structure of Application Integration Development must compliant with General Folder Structure. Other directories can be added depends on project requirements.

In folder “src”, creates an individual sub-folder for each module in which contains C code file and internal header files. The header files (in most of the cases just one head file for each module) which define interfaces and open data structure definition or objects must settle on public “include” folder of such project. The project only has one public “include” folder which contains the open header files of all the modules.

* 1. Library Development

The Library Development Project contains folders: “build”, “src”, “include”, “lib”, “doc”. The structure of folder “build” and “doc” are compliant with General Folder Structure.

The folder “src” contains all the C source code files and internal header files of each module.

The folder “include” contains all the public header files of each module.

The folder “lib” contains all the compiled dynamic link library files on every possible supported platform. The DLL files of every platform settle on individual sub-folder with platform name which must be standardized as: “linux”, “win32”, “macos”, “vxWorks”.

* 1. Module Development

The Module Development Project contains folder: “build”, “src”, “doc”.

1. Name Regulation

The module name which generally represents as the module folder name must be identical with module open header file.