Vapula编码规范

版本：2013/12/12（重制2修订6）

*此文档介绍Vapula框架开发需要遵循的编码规范。*

1. **命名空间约束**

C/C++代码需要使用vapula命名空间进行限定。

.NET代码使用Vapula命名空间进行限定。

1. **命名规范**

Vapula的所有语言的常量标识符全部大写，使用下划线分隔单词，

VF（Vapula Framework）作为前缀。

Vapula C/C++代码的C导出函数使用vfe（Vapula Framework Entry）作为函数名前缀。

禁止在任何场合使用匈牙利命名法（e.g. iCount），

不能添加以下辅助标示符号：类前缀C、成员变量前缀m、指针前缀ptr。

类成员的标识符使用Pascal命名法（e.g. GetUser），

私有成员前缀下划线，与其他访问级别加以区分（e.g. \_DataSource）。

函数形参尽量简化，原则上不要超过一个单词，

超过一个单词使用小驼峰命名法（e.g. dataSource），

或下划线命名法（e.g. max\_recv\_length）。

临时变量必须使用下划线命名法（e.g. max\_recv\_length）。

布尔型命名需要指定Is或Has或Can或Able等。

1. **注释规范**

注释要求简洁、易懂。

注释可以用来说明：代码用途，算法或代码问题，改进的目标。

注释不能包含任何自动化API文档抽取标示。

注释不要包含代码，如有代码不启用且暂不删除，请移植到临时文件。

注释不要说明详细的算法原理，开发人员应当通过算法文档了解算法。

注释不要包含任何可能与开发人员个人隐私相关的信息，

包括开发人员昵称、代码修改时间、版权等。

以最少注释为根本原则。

如果必须使用长句，不能将长句分行。

注释如果违背语句规范，不得保持超过2个提交。

1. **代码块编写**

一个函数不要超过100行。

不限定括号闭合风格。

一个函数原则上只能具备一个功能，

避免使用开关变量选择功能。

避免使用纷繁复杂的break或continue

避免使用超过3层嵌套的条件判断。

避免使用内联函数。

1. **语句编写**

每一行不要超过160字符。

不要在代码中出现Unicode字符。