TCM范式与组件化开发

版本：2013/12/9（重制0修订1）

*此文档介绍TCM范式。*

绿色表示不准确，黑色为确定。

TCM范式是一系列软件开发的规则集合。

1. 目标软件产品包含两部分：主应用程序框架和业务组件集。
2. 业务组件集和主应用程序之间的通信逻辑应当独立于主应用程序框架和业务组件集。
3. 无论开发或运行，业务组件集与主应用程序应当互相独立，不互相影响。
4. 业务组件集的组件之间也应当互相独立。
5. 组件与组件的通信通过中间层基于消息机制实现。
6. 组件与平台可以分别测试。
7. 组件脱离平台可执行。
8. 组件可以通过TCM流程图描述并构建独立的业务应用。
9. 组件可以跨运行时集成。
10. 组件框架应当支持运行时依赖的按需加载。
11. 组件的UI可以跨技术风格开发。
12. 组件的UI通过Dispatcher进行多进程UI承载。
13. 任务指一次指定组件指定功能指定参数的调用
14. 组件结合TCM流程图构成一个业务模型。
15. TCM范式框架的项目命名和命名空间不能使用“TCM”。
16. TCM范式框架的函数命名应当避免包含“TCM”。
17. 一台设备上可以部署多种TCM范式的框架实现。

## TCM设计

TCM是一种高性能、低资源占用的组件化软件开发技术，

支持跨语言开发，能够跨平台运行，支持组件操作注入，支持任务式调用。

字面上，TCM指Task（任务）+Component（组件）+Model（模型）。

其中表达了这样的涵义：

功能可被归类在组件中，组件调用可被抽象为任务，组件可以参与复杂建模。

在软件项目的整个开发周期中应用TCM技术，

能够显著提高开发效率，有效降低架构耦合程度和项目复杂度。

以下简述TCM技术的关键机制，具体请参考其他有关TCM设计的文档。

驱动机制 与 跨语言和跨平台特性

开发人员可以根据自己的需要，选择任何支持的语言开发TCM组件，

任何TCM组件可以用于任何一种支持的执行环境。

开发人员也可以按照TCM的相关规范，开发用于所需特定语言和平台的驱动。

插件机制 与 组件操作注入特性

“组件操作注入”指开发人员在组件的功能的执行过程中将某些对象与组件的执行过程关联。

TCM遵循AOP思想，提供了特性插件机制。

开发人员可以根据自己的需要，在运行期间为执行过程注入特性对象。

特性对象包括日志、计时器等。

宿主机制 与 任务式组件调用特性

开发人员可以通过任务式调用隔离应用与组件。

在保证效率和功能不损失的前提下，进一步提高应用的稳定性，

并可以适用于更多样复杂的集成场景。