用于 TCM（Task Component Model）的平台开发规范
版本：2013/1/17（修订 7）

1.

TCM 是一套中间件，用于简化组件开发，支持多种开发语言。

目前，TCM 仅支持 Windows XP 以上操作系统。

在使用 TCM 组件之前，建议开发人员为平台准备一个描述文档。

在文档中，应当记录以下规范中要求的事项描述。

2.

组件的物理形式是一系列文件的集合。

包括：

[组件标识].dll（标准 C++动态链接库，×1）

[组件标识].tcm.xml（组件配置文件，×1）

[组件标识].[功能标识].tcm.png（组件图标组，×m）

[组件标识].[附属程序集标识].dll/exe（组件附属程序集，×n）

组件名称和附属程序集名称仅限于字符集合[A-Za-z0-9_]。

.dll 和.exe 以及由组件管理器或组件发布器生成的 tcmcoms.xml（组件清单文件）是调用 TCM 组件必须的。

组件配置文件则用于发布组件和安装组件。

3.

开发人员可以自行开发组件管理器。

具体的管理策略请参考附录（附录尚未整理，请搁置）。

如果不打算自行开发组件管理器，可以通过 Shell 调用 tcm_ commng.exe（SDK/Toolkits）。

目前，组件管理器尚未完成开发。

关于组件的规范，请参考《TCM 组件开发规范》。

4.

tcm_bridge（以下称为 TCM 桥）中提供了供组件与平台通讯所需的：

参数信封对象、上下文对象和执行器对象。

无论选择什么语言开发平台，在使用 TCM 组件时，都需要通过 TCM 桥作为中间层。

使用.NET 时，TCM.NET 运行时会自动通过 TCM 桥进行组件调用。

使用 C/C++时，请引用头文件“tcm_bridge.h”，并链接相关输入。

使用其他语言时，请通过 Win32 DLL 动态加载机制获取 TCM 桥提供的纯 C 接口。

TCM 桥提供的纯 C 接口可以执行几乎所有 TCM 桥支持的任务。

TCM 组件的建议用途是执行无交互的自动化任务，所以 TCM 组件中通常不应具有用户界面。

平台应该自己实现用户界面策略。

需要平台开发者注意的是，TCM 支持多种参数类型，建议用户界面最好是在运行时动态生成的。特殊的、专用的用户交互过程不在讨论范围。

平台不应该自行实现对 TCM 组件的调用细节。

5.

一个组件一定包含以下 C 风格导出函数。

1) 主调函数

主调函数内，实现且仅实现功能的跳转（或调用），具体跳转流程参见组件文档。

入口点签名：*UINT Run(int function, tcmEnvelope* envelope, tcmContext* context)*

关于输入参数

function 是功能的标识。

envelope 是用于参数交换的信封对象。关于信封对象的能力详见后文。

context 是当前功能的执行上下文。关于上下文对象的能力详见后文。

关于返回值

返回值是用于描述功能调用状态的通用信息，不会用来作为功能的输出参数。

如果用户希望返回**关乎组件功能细节**的信息，这些信息会作为**参数**装填在信封中传递。

以下是 TCM 为组件入口点的返回值定义的含义：

返回值	含义
0	未定义的错误
1	正常结束
2	用户取消，通知方式
3	用户取消，强制方式
4	功能不存在
[5,DWORD_MAX]	保留，用于 TCM 未来的扩展定义

6.

C++平台可以使用上下文对象（tcmContext）操作执行状态。

通过上下文对象，平台可以发布控制码（CtrlCode），也可以监听组件广播的进度更新。

注意，控制码本身只是来自平台的通知，控制码没有强制能力。一般，组件自己会监听控制码并进行响应动作。

上下文对象中定义了 3 个控制码，如下表。

标识符	含义
TCM_CTRL_CANCEL	要求取消
TCM_CTRL_PAUSE	要求暂停
TCM_CTRL_RESUME	要求恢复

如果平台自己实现调用细节，需要保证功能的状态和返回值是记录完备的。

即：平台在实现调用细节时，应当在合适的时机根据组件的反馈写入状态码和返回值。

上下文对象中定义了 3 个状态码，如下表。

标识符	含义
TCM_STATE_IDLE	空闲
TCM_STATE_RUNNING	执行中
TCM_ STATE_PAUSING	暂停中

上下文对象面向平台开发提供以下常用方法。

签名	说明
void SetState(int state)	设置状态
int GetState()	获取状态
void SetReturnCode(int retcode)	设置返回值
int GetReturnCode()	获取返回值
void SwitchPause()	切换真实执行状态的运行和暂停
int GetCtrlCode()	获取控制码
void CallCancel()	发布取消控制码
void CallPause()	发布暂停控制码
void CallResume()	发布恢复控制码
float GetProgress()	获取当前执行进度
void SetProgress(float value)	设置当前执行进度 请为 value 传入[0,100]的浮点数

上下文对象是线程安全的。

7.

C++平台使用信封对象（tcmEnvelope）与组件进行参数通信。

一般请在调用执行器前构造信封对象。

信封对象提供以下常用构造或创建方法。

签名	说明
tcmEnvelope (int total, int* types, bool* ins)	构造函数（已知参数总数，类型表和方向表）
static tcmEnvelope* Parse(PCSTR xmlstr)	工厂方法（已知参数描述 XML 的字符串）
tcmEnvelope* tcmLoadParams (PCWSTR pathlst, PCSTR cid, int fid)	工厂方法 （已知组件清单路径，组件标识和功能标识）

建议使用工厂方法，第二种工厂方法效率较低。

TCM 支持**全部**数据类型，但不支持**任意**类型，具体见下表。

类型标识符	描述
TCM_DT_POINTER	指针
TCM_DT_INT8	8 位整数
TCM_DT_UINT8	无符号 8 位整数
TCM_DT_INT16	16 位整数
TCM_DT_UINT16	无符号 16 位整数
TCM_DT_INT32	32 位整数
TCM_DT_UINT32	无符号 32 位整数
TCM_DT_INT64	64 位整数
TCM_DT_UINT64	无符号 64 位整数
TCM_DT_REAL32	32 位浮点数
TCM_DT_REAL64	64 位浮点数
TCM_DT_BOOL	布尔值
TCM_DT_CSTRA	多字节常字符串
TCM_DT_CSTRW	宽字节常字符串

用户对象和数组应当使用指针替代。

随着 TCM 的更新，内存管理机制将会愈加完善。

向信封对象写入用户对象时，建议同时提供该对象的实际大小。

当开发人员提供对象大小（非 0）时，TCM 将执行**浅拷贝**，在内存中生成对应对象的浅表副本，并记录此指针。这样，即便原有对象由于某些原因失效，写入信封中的对象依然有效。

如果开发人员不提供对象的实际大小（0），TCM 将仅写入引用。

这种情况下请自行保证该对象的生命周期。

TCM 提供转型读写支持。

信封对象面向组件开发人员提供以下常用方法。

签名	说明
int GetParamTotal()	获取参数数量
bool GetInState(int id)	获取参数方向
T Read(int id)	读取参数
void Write(int id, T value)	写入参数
void Write(int id, LPVOID value, int size)	写入对象（重载）
void Write(int id, PCSTR value)	写入多字节常字符串（重载）
void Write(int id, PCWSTR value)	写入宽字节常字符串（重载）
PCSTR CastReadA(int id)	读出数值并自动转型到字符串
PCWSTR CastReadW(int id)	读出数值并自动转型到字符串（Unicode）
void CastWriteA(int id, PCSTR value)	自动转型到数值并写入
void CastWriteW(int id, PCWSTR value)	自动转型到数值并写入（Unicode）
void Deliver(tcmEnvelope* who, int from, int to)	投递当前信封到目标信封
void CastDeliver (tcmEnvelope* who, int from, int to)	投递当前信封到目标信封 投递时自动匹配类型（效率较低）

8.

C++平台使用执行器对象（tcmExecutor）调用 TCM 组件。

使用执行器完整执行一个组件的一个功能需要以下几个步骤：

1) 根据组件清单获得信封对象
2) 加载组件 DLL
3) 构造一个执行器
4) 装载执行器
5) 构造一个上下文对象
6) 启动执行器
7) 等待功能异步执行完成。期间可以上下文检测进度更新通知、暂停、恢复、终止功能。
8) 功能执行完成，返回
9) 组件检测到功能执行完成。此时可以使用功能的参数信封、上下文中的返回值等。
10) 卸载执行器
11) 卸载组件 DLL（可选）

执行器对象提供以下构造方法

签名	说明
tcmExecutor(HMODULE module)	构造函数（已知组件 DLL 句柄）

执行器对象面向平台开发人员提供以下常用方法。

签名	说明
void Mount(tcmEnvelope* env, int fid)	为执行装载功能
void Unmount()	卸载执行器
bool Start(tcmContext* ctx)	启动功能（已获得上下文对象）
void Stop(int wait = 0)	终止功能（已知同步等待时间）

9.

其他常用函数

TCM 为组件开发人员提供了一系列工具函数。

签名	说明
PCWSTR tcmMbToWc(PCSTR src,UINT codepage)	转换多字节字符串到宽字节字符串
PCSTR tcmWcToMb(PCWSTR src,UINT codepage)	转换宽字节字符串到多字节字符串
PCSTR tcmValueToStrA(T value)	转换数值到字符串
PCWSTR tcmValueToStrW(T value)	转换数值到字符串（Unicode）
T* tcmVectorToArray(vector<T>& src)	转换 vector 容器到定长（堆）数组
int tcmGetTypeUnit(int type)	获得 TCM 类型的单位大小
void tcmAssert(bool state,LPVOID clear)	根据断言结果执行清理
PCSTR tcmCopyStrA(PCSTR src)	复制一个字符串
PCWSTR tcmCopyStrW(PCWSTR src)	复制一个字符串（Unicode）
PCSTR tcmGetRandomStrA(int len)	获得一个随机 HEX 字符串
PCWSTR tcmGetRandomStrW(int len)	获得一个随机 HEX 字符串（Unicode）