以下关于TCM应用的例子中，组件代码中需要添加TCM Bridge要求的C接口。

**应用场景一：进度**

有一段代码，运行时间较长，用户需要知道进度，怎么办？

1. 开发人员自行计算进度，在代码中通过tcm::Context类提交进度
2. 调用方使用TCM Bridge获得进度

**应用场景二：运行控制**

有一段代码，运行期间，用户想要暂停/恢复或终止，怎么办？

1. 开发人员通过tcm::Context类在合理的时机监听指令，

并提供一段代码进行对应的控制操作

1. 调用方使用TCM Bridge发布控制指令即可

**应用场景三：异常反馈**

有一段代码，运行终止时（正常或失败），用户想要知道发生的异常，怎么办？

1. 开发人员通过tcm::Context类在发生异常的时刻 提交 希望反馈的异常内容
2. 调用方使用TCM Bridge获得异常信息

**应用场景四：任务单**

有一个软件，集成方要求使用（专用的）任务单来调用组件，

团队不希望 添加专用任务单解析的代码 污染 已有的功能代码（因为专用代码会降低代码复用能力）、还要能够响应任务单，怎么办？

1. 在组件中快速简单地添加TCM调用支持
2. 使用TCM Host进行组件调用
3. 如果集成方 不接受 团队的任务单规范，团队方启用TCM Host的脚本扩展模式，并提供转换脚本将集成方的任务单翻译成TCM Host任务单规范。

**应用场景五：组件通信规范**

有一个软件，集成方 要求使用（专用的）通信规范来获得组件的进度、状态、错误反馈，

团队不希望 添加专用组件通信的代码 污染 已有的功能代码（因为专用代码会降低代码复用能力）、还要能够响应任务单，怎么办？

1. 在组件中快速简单地添加TCM调用支持
2. 使用TCM Host进行组件调用
3. 启用TCM Host的脚本扩展模式，并提供通信脚本以 适应 集成方的规范要求

（脚本的开发和调试成本大大低于功能的代码）

**应用场景六：工作流**

有一个软件，细分功能很多，团队希望 向用户开放 将细分功能组织成工作流的特性，又不想专门开发工作流软件。怎么办？

1. 让所有细分功能支持TCM Bridge
2. 使用TCM ModelDesigner（该产品已搁置）

**应用场景七：跨进程**

有一个软件，有许多功能，很多功能没有充分的测试和回归时间，使用中很容易崩溃，并导致整个软件崩溃。团队希望在现有条件下一劳永逸解决问题，怎么办？

1. 让功能支持TCM Bridge
2. 使用TCM Host进行进程隔离托管运行
3. 当功能发生问题时，仅对应功能的宿主进程会崩溃，不影响整个软件

**应用场景八：跨语言调用**

有一些软件模块

（使用C/C++、C#、VB.NET、Java、IDL、Python、Matlab、Lua、javascript开发），

用户希望使用多种语言调用

（使用C/C++、C#、VB.NET、Java、IDL、Python、Matlab、Lua、javascript调用），

怎么办？

1. 让功能支持TCM Bridge（目前仅C/C++支持）
2. 如果调用方是C/C++、C#、VB.NET，可以直接使用TCM Bridge调用
3. 如果调用方是Python、Lua，可以使用TCM Bridge的C接口调用（已实现，未测试）
4. Java、Matlab、javascript暂缓实现

**应用场景九：分布式调用**

有一些软件模块，用户希望自建一个小规模工作站网络，并在其上分布式调用模块、监测模块运行状况、控制模块运行，怎么办？

1. 让功能支持TCM Bridge
2. 使用TCM Server进行分布式网络服务（暂列，未开发）

**应用场景十：组件管理**

有一个软件，功能模块很多，团队只想快速开发，不想再分出精力管理以及向用户推送更新，怎么办？

1. 让功能模块支持TCM Bridge
2. 使用TCM Publish维护版本、推送更新（暂列，未开发）