



IBM China Development Lab

WebSphere Process Server V7

调用样式和事务设置介绍

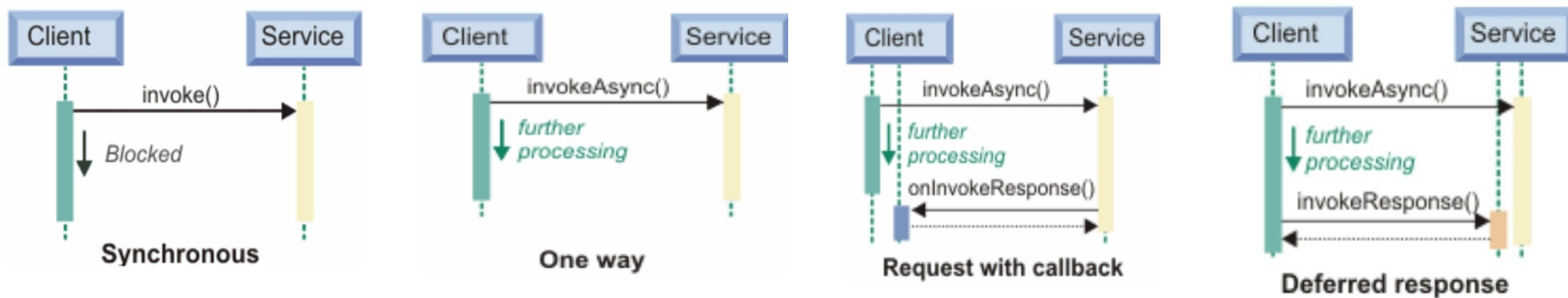
高明
WPS 二线技术支持

目录

- 调用样式介绍
- 调用样式调优
- 事务设置介绍
- 事务设置调优
- 总结
- Q&A

WPS 调用样式介绍 – 调用样式定义及分类

- 调用样式定义
 - 调用样式决定了 SCA 组件之间以什么方式来进行通信
- 调用样式分类
 - 同步调用
 - 单向的（One way）异步调用
 - 请求回调（request with callback）方式的异步调用
 - 延迟响应（deferred response）方式的异步调用
- 同步异步调用的实现方式

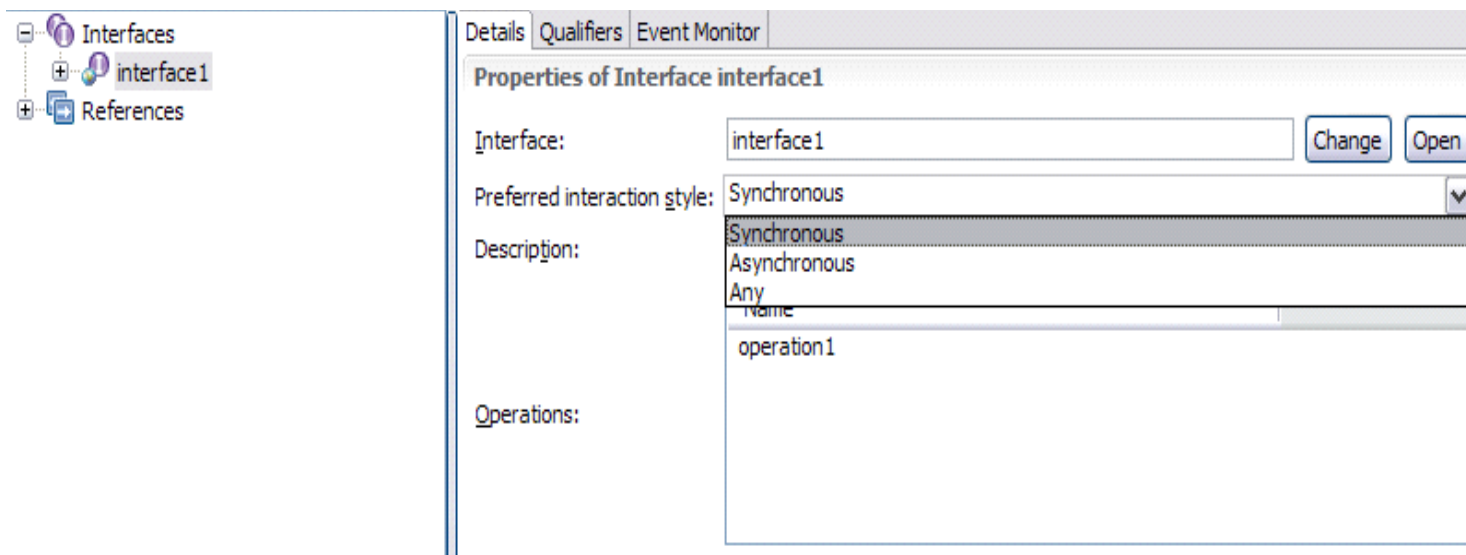


WPS 调用样式介绍 – 调用样式的决定因素

- 调用样式的决定因素
 - 目标 SCA 接口端设置 Preferred Interaction Style
 - 调用发起端的组件
 - Java 组件
 - BPEL 组件（长流程，短流程）
 - 调制流（Mediation Flow）
 - JMS/MQ/MQJMS 绑定导入
 - HTTP 和 Web Service 绑定导入
 - SCA 导出
 - 接口映射
业务规则和选择器
 - 接口操作类型：单向调用（one way） / 双向调用（two way）

WPS 调用样式介绍 - 设置方式

- 目标 SCA 接口端设置的 Preferred Interaction Style
 - 在每个 SCA 组件的接口上设置
 - 可以选择同步、异步和任意
 - 只能影响到最终实际发生的调用样式
 - 不能完全决定最终实际发生的调用样式



WPS 调用样式介绍

- 不同类型 SCA 组件的行为方式
 - Java 组件：Java 组件是通过 Java 代码控制最终的调用方式
 - invoke()：用于实现同步调用
 - invokeAsync()：用于实现单向异步调用和延迟响应异步调用
 - invokeAsyncWithCallback()：用于实现请求回调式异步调用
 - BPEL 长流程
 - 决定因素
 - 目标组件接口的 Preferred Interaction Style
 - 接口操作类型

长流程调用样式表

目标组件或导入的 Preferred Interaction Style	单向调用 (One-way operation)	请求响应调用 (Request-response operation)
任意	异步调用	异步调用
同步	同步调用	同步调用
异步	异步调用	异步调用

WPS 调用样式介绍

- 不同类型 SCA 组件的行为方式
 - BPEL 短流程
 - 决定因素
 - 目标组件接口的 Preferred Interaction Style
 - 接口操作类型

短流程调用样式表

目标组件或导入的 Preferred Interaction Style	单向调用 (One-way operation)	请求响应调用 (Request-response operation)
任意	异步调用	同步调用
同步	同步调用	同步调用
异步	异步调用	同步调用

WPS 调用样式介绍

- 不同类型 SCA 组件的行为方式
 - 调制流 (Mediation Flow)
 - 两种对外调用的 primitive
 - Callout
 - Service Invoke
 - Invocation Style 设置
 - Require mediation flow to wait for service response when the flow component is invoked asynchronously with callback 设置

Require mediation flow to wait for service response when the flow component is invoked asynchronously with callback.

Invocation Style:

Sync	▼
Default	
Sync	
Async	

WPS 调用样式介绍

- 不同类型 SCA 组件的行为方式
 - 调制流 (Mediation Flow)
 - 两种对外调用的 primitive
 - Callout
 - Service Invoke
 - Invocation Style 设置
 - Require mediation flow to wait for service response when the flow component is invoked asynchronously with callback 设置

Require mediation flow to wait for service response when the flow component is invoked asynchronously with callback.

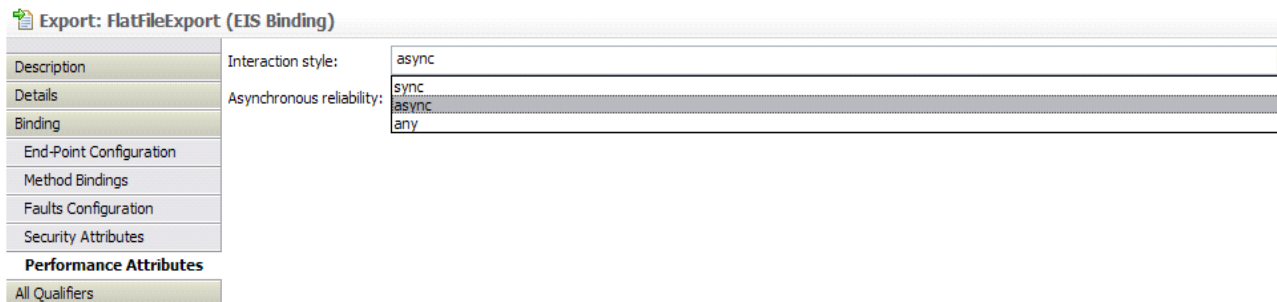
Invocation Style:

Sync	▼
Default	
Sync	
Async	

调制流如何被调用	目标组件的 Preferred Interaction Style	操作类型	Require mediation flow to wait for service response when the flow component is invoked asynchronously with callback	Callout 的调用样式
同步	任意	单向调用	true 或 false	异步
	任意	请求响应	true 或 false	同步
	同步	任意	true 或 false	同步
	异步	任意	true 或 false	异步
延迟响应异步	任意	单向	true 或 false	异步
	任意	请求响应	true 或 false	同步
	同步	任意	true 或 false	同步
	异步	任意	true 或 false	异步
请求回调异步	任意	单向	true 或 false	异步
	任意	请求响应	true	异步
	任意	请求响应	false	请求回调异步
	同步	任意	true 或 false	同步
	异步	单向	true 或 false	异步
	异步	请求响应	true	异步
异步	请求响应	false	请求回调异步	

WPS 调用样式介绍

- 不同类型 SCA 组件的行为方式
 - JMS/MQ/MQJMS 绑定导入
 - 全部异步调用目标服务
 - HTTP 和 Web Service 绑定导入
 - 全部同步调用目标服务
 - SCA 导出
 - EIS 导出
 - 调用样式由 Performance Attributes -> Interaction Style 决定



- JMS/MQ/MQJMS 绑定导出、SCA JMS 绑定导出、HTTP 绑定导出和 WebService 绑定导出

WPS 调用样式介绍

EIS 调用样式表

调用方式	性能属性的交互样式	调用样式
单向调用	异步	异步调用
单向调用	同步	同步调用
请求响应	任意	同步调用

导出调用样式表

导出类型	操作类型	调用样式
MQ 或 MQ JMS	单向	异步调用
SCA JMS	单向	异步调用
SCA JMS	请求响应	异步调用（请求回调）
Web Service（soap over http/jms）	单向或请求响应	同步调用

WPS 调用样式介绍

- 不同类型 SCA 组件的行为方式
 - 接口映射
 - 在 WPS V7 中接口映射已经是不被推荐使用的了
 - 使用调制流（Mediation Flow）得到同样的功能

接口映射调用样式表

传入调用样式	客户端操作类型	目标操作类型	目标组件或导入的 Preferred Interaction Style	调用样式
同步	任意	任意	同步	同步
	单向	请求响应	异步	异步（请求回调）
	请求响应	任意	异步	同步
	任意	任意	任意	同步
异步	任意	任意	同步	同步
	任意	单向	异步	异步
	任意	请求响应	异步	异步（请求回调）
	任意	单向	任意	异步
	任意	请求响应	任意	异步（请求回调）

WPS 调用样式介绍

- 不同类型 SCA 组件的行为方式
 - 业务规则和选择器
 - 业务规则和选择器始终同步调用目标端
 - 如果业务规则和选择器选择的目标组件接口的 Preferred Interaction Style 设为异步，则有可能会出现异步转同步。

WPS 调用样式性能调优

- 谨慎使用异步调用
 - JMS 或 MQ 等消息通信进行异步调用，会消耗更多资源
- 尽量将 Preferred Interaction Style 设置为同步
 - 跨组件的同步调用能够提供较好的性能
 - 注意事项
 - Preferred Interaction Style 的继承性
 - 对于 SCA 绑定导入导出，导入导出的调用样式是由 SCA 导入接口的 Preferred Interaction Style 设置决定的。
 - 对于肯定为异步调用的绑定导入，如 JMS/MQ 绑定导入，可以不用设置其接口的 Preferred Interaction Style 为异步（任意 Any 即可），这样可以减少 WPS 运行时的额外处理，提高性能。
- 避免在 FanOut/FanIn 中使用异步方式调用同步服务
 - 同步异步调用选择的利弊
 - 异步调用耗费更多资源
 - 同步调用的耗费时间有可能更长
- 避免双向（two way）的同步转异步调用
 - 反模式（anti-pattern）：同步转异步



WPS 事务设置介绍

■ 事务设置定义

- 事务是一组活动单元，在一个事务中，一组对于资源的更新活动将以原子的方式执行。进一步说，要么事务中所有的资源更新都被执行并被永久保存，要么没有任何资源更新被永久保存。

■ 事务设置分类

- SCA 组件
- BPEL 组件
- 异步调用中的特殊事务设置

WPS 事务设置介绍 – SCA 组件级别的事务

- 接口（Interface）服务质量限定符
 - Join transaction：决定了目标服务是否有能力加入到客户端（client）发起的事务中
 - True：服务组件加入到客户端传播过来的事务中
 - False：服务组件不加入到客户端传播过来的事务中
 - Join transaction 设为 True 生效的条件
 - 调用端发起的必须是同步调用。异步调用是不能将目标服务加入到事务中的。也就是说当客户端使用异步调用的方式发起调用，Join transaction 限定词将被忽略，不会加入客户端的事务中
 - 目标服务组件实现中的 Transaction 限定符必须设为 Global

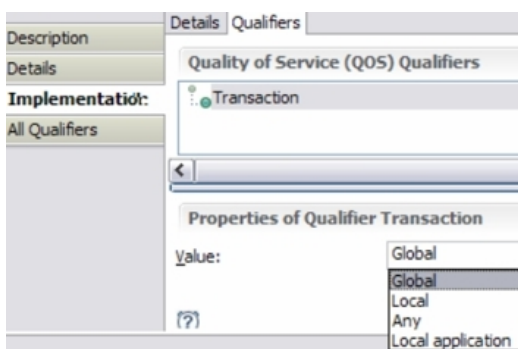


WPS 事务设置介绍 - SCA 组件级别的事务

■ 实现 (Implementations) 服务质量限定符

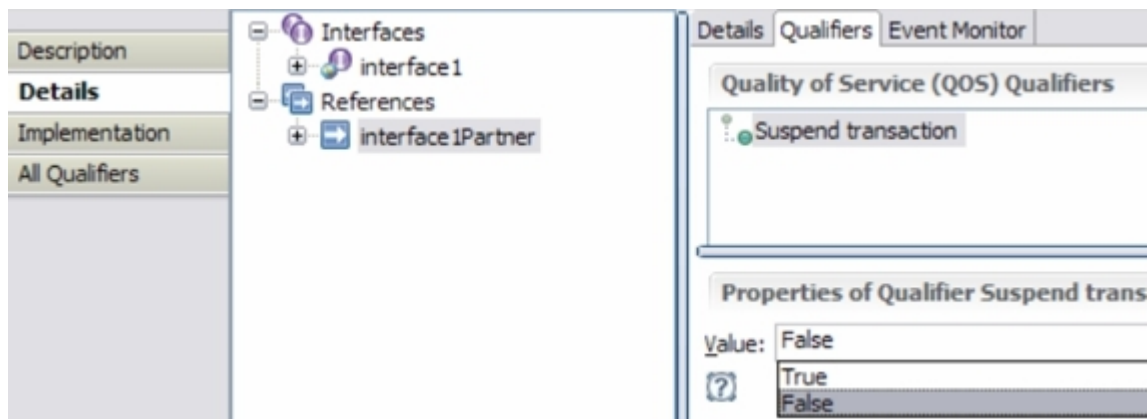
- Transaction: 决定服务组件功能单元的逻辑划分

- Global: 若客户端发起并传播了全局事务 (Global transaction), 并且目标组件 (SCA 组件) 设置为加入事务 (Join transaction 为 True), 那么该组件将加入到客户端发起的全局事务中。如果不满足上述条件, 则目标组件将为自己新建一个全局事务。
- Local: 如果目标组件不支持两阶段提交, 则需要设置为 Local。当设为 Local 时, 目标组件将运行于自己的局部事务中, 既不会加入到客户端的事务中, 也不会让被它调用的服务组件加入到自己的局部事务中。
- Any: 若客户端传播过来全局事务上下文 (global transaction context), 则服务组件加入到全局事务中。否则目标组件将创建自己的局部事务。
- Local Application: 当目标组件运行于局部事务容器 (Local transaction containment), 且局部事务容器由应用程序负责管理时选择此选项。换句话说, 当设为 Local Application 时, 相当于 Transaction 服务质量限定词被设为不支持, 并且 LocalTranResolver 设为 application。事务由应用程序自己负责管理提交或回滚。



WPS 事务设置介绍 - SCA 组件级别的事务

- 引用（References）服务质量限定符
 - Suspend transaction：当服务组件本身作为客户端同步调用目标组件时，此选项决定客户端是否将事务传播给被调用的目标组件。
 - False：客户端的全局事务向目标组件传播，目标组件与当前服务组件（客户端）在一个全局事务中。
 - True：客户端的全局事务不会向目标组件传播，目标组件将运行于自己的全局事务中或局部事务中。
 - Suspend transaction 限定词被忽略的条件
 - 当前服务组件使用的是局部事务
 - 异步调用目标服务组件



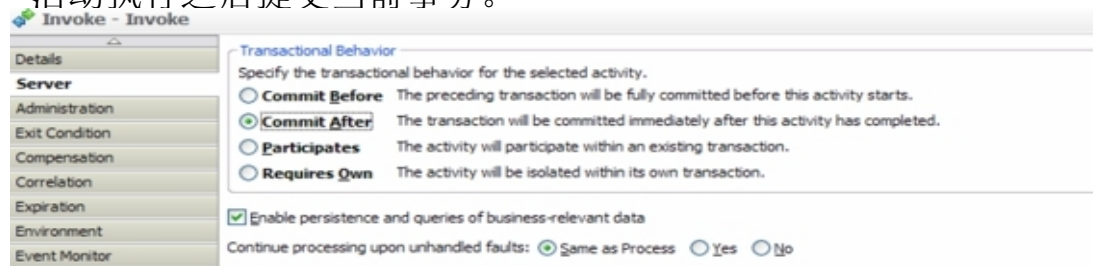
WPS 事务设置介绍 - 业务流程组件级别的事务

■ 业务流程分类

- 短流程：短流程用于短时间存在的流程，短流程总是存在于一个事务中，状态记录于内存中，而不会记录到数据库中。
- 长流程：长流程相对于短流程而言其存活时间非常长，可以运行数小时到数天甚至数年的时间。长流程通常跨越多个事务，每个事务通过 JMS 或 Work Manager 发起。

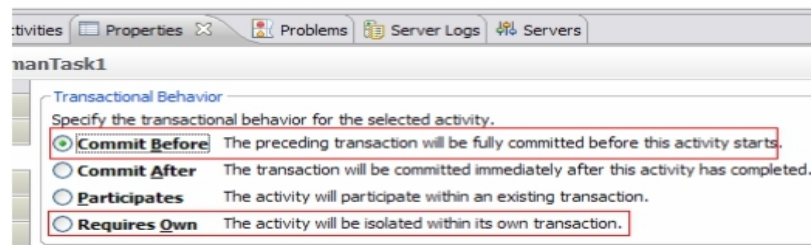
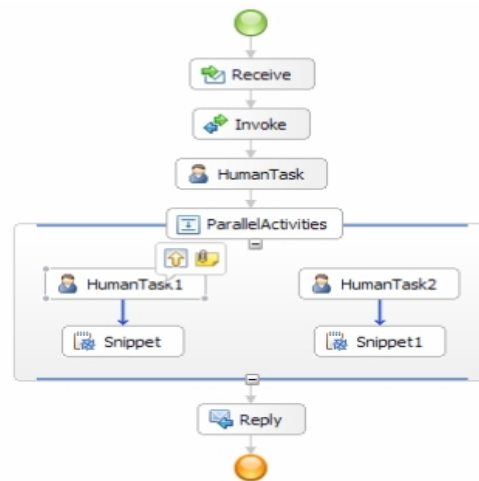
■ 活动事务设置

- Commit before：之前的事务结束，当前的活动启动一个新的事务。
- Commit after：当前活动加入到之前的事务中，并且在此活动结束时提交事务，下一个跟着的活动会重启一个新事务。
- Participates：当前的活动加入到之前的事务中，并且在此活动结束后是否提交事务取决当前活动的种类以及于下一个活动的事务行为设置。
- Requires own：之前的事务结束，当前的活动启动一个新的事务，并且在当前活动执行之后提交当前事务。



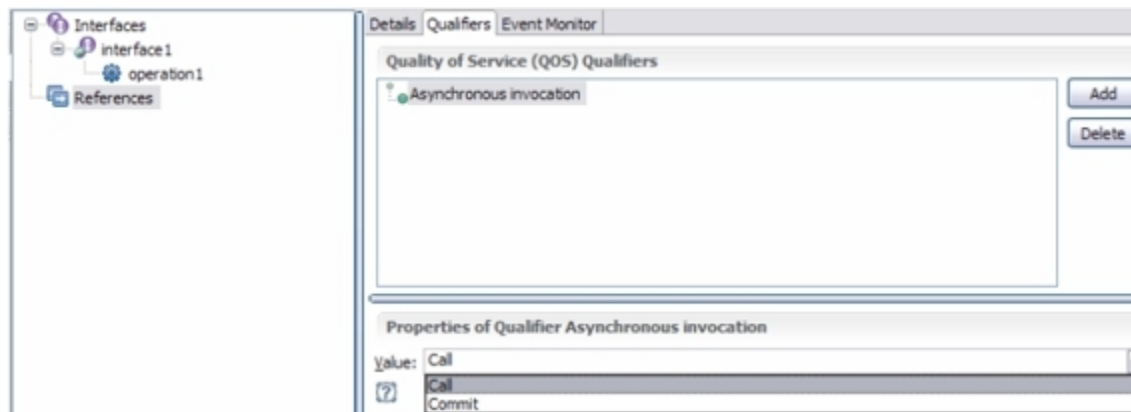
WPS 事务设置介绍 - 业务流程组件级别的事务

- 平行分支（parallel branches）并行导航的事务设置
 - 为了达到每个分支的并发性，每个分支都要在开始点启动一个自己的事务，也就是说每个分支的第一个活动的行为应该设为 Commit before 或 Requires own。



WPS 事务设置介绍 – 异步调用中的事务设置

- 引用（References）服务质量限定符 – Asynchronous invocation
 - SCA 组件异步调用的两个阶段
 - 进行自身的业务处理
 - 发起异步调用（向队列发送消息）
 - Asynchronous invocation 用于设置消息发送是否加入客户端自身的事务中
 - 两个选项
 - Call：发送消息独立于客户端自身的事务
 - Commit：送消息与客户端自身存在于一个事务中
 - 谨慎选择“commit”选项



WPS 事务设置性能调优

- SCA 组件的事务设置调优 – 通常一个事务的执行速度会快于多个事务
 - 对于客户端（调用发起者），将引用服务质量的 `SuspendTransaction` 限制符设为 `false`。
 - 对于被调用者，将接口服务质量 `joinTransaction` 限制符设为 `true`。
 - 对于每个 SCA 组件的实现服务质量 `Transaction` 限定符设为 `any` 或 `global`。
- 工作流中事务的调优
 - 设置 `Participates` 可以使长流程导航拥有最好性能
 - 例外
 - 平行分支（`parallel branches`）的并行导航应设为 `Commit before` 或 `Requires own`
 - `inline` 类型的人工任务，使用 `Commit after` 可以得到最优人工任务查询性能
- 避免双向（`two way`）的同步转异步调用
 - 反模式（`anti-pattern`）：同步转异步



总结

- 调用样式和事务设置是 **WPS** 中重要的基本概念和功能
- 正确设置调用样式和事务可以完成更复杂需求，提高程序性能
- 设置原则：
 - 首先满足功能需求
 - 其次考虑达到性能最优
 - 避免 anti-pattern：同步转异步

Q&A

参考文档

- developerWorks :

WebSphere Process Server V7 性能调优, 第 1 部分: 调用样式

http://www.ibm.com/developerworks/cn/websphere/library/techarticles/1107_gaom_wpstuning/1107_gaom_wpstuning.html

WebSphere Process Server V7 性能调优, 第 2 部分: 事务设置

http://www.ibm.com/developerworks/cn/websphere/library/techarticles/1107_gaom_wpstuningPart2/1107_gaom_wpstuningPart2.html

谢谢！！