

版本 6.1.0



业务流程编排器



业务流程编排器

注意

在使用本资料前，请务必阅读本文档末尾声明一节中的一般信息。

2008 年 2 月 1 日

此版本适用于 WebSphere Process Server for Multiplatforms V6.1.0（产品编号为 5724-L01）及所有后续发行版和修订版，直到在新版本中另有声明为止。

要向我们发送您对本文档的意见，请发送电子邮件消息到 doc-comments@us.ibm.com。我们期待您的宝贵建议。

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 对于您所提供的任何信息，有权利以任何它认为适当的方式使用或分发，而不必对您负任何责任。

© Copyright International Business Machines Corporation 2006, 2008. All rights reserved.

目录

第 1 部分 WebSphere Process Server 中的业务流程和人员任务 1

第 1 章 关于业务流程 3

流程模板	4
业务流程类型	5
流程版本控制	5
流程实例	6
流程生命周期	6
流程实例的状态过渡图	6
子流程的生命周期管理	9
活动的状态过渡图	9
业务流程的调用方案	15
业务流程交互的影响因素	15
在业务流程与服务之间传递参数	16
业务流程的事务行为	17
微流程的事务行为	17
长时间运行流程的事务行为	19
业务流程中的故障处理和补偿处理	22
故障处理	22
业务流程中的补偿	25
从基础结构故障恢复	26
流程的授权与人员分配	28
业务流程的授权角色	28
用于创建和启动业务流程的授权	30
用于管理业务流程的授权	31

第 2 章 关于人员任务 33

任务模板	34
人员任务的类型	35
对人员任务进行版本控制	36
任务实例	36
独立任务与直接插入任务	36
子任务	39
后续任务	41
升级	43
升级的电子邮件通知	45
人员任务的生命周期	45
任务调用方案	49
影响独立调用任务及其服务组件的行为的因素	51
方案: 支持以异步方式调用服务的独立调用任务	52
方案: 支持以异步方式和同步方式调用服务的独立调用任务	54
授权与人员分配	56
人员任务的授权角色	56
授权与工作项	58
人员分配条件	59
人员解析	68
缺席者替代	74
缺省人员分配	74

管理人员分配条件和人员解析结果	76
共享人员分配	77

第 2 部分 规划和配置业务流程编排器 79

第 3 章 规划配置业务流程编排器 81

规划拓扑、设置和配置路径	81
规划创建业务流程编排器基本样本配置	84
规划创建包含样本公司的业务流程编排器样本配置	85
规划非生产部署环境配置	85
规划使用管理控制台的部署环境向导	86
规划定制业务流程编排器配置	89
规划安全性、用户标识和权限	89
规划业务流程编排器的数据库	95
规划业务流程管理器和人员任务管理器	105
规划人员目录提供程序	106
规划业务流程编排器资源管理器	107
规划业务流程编排器观察器	108
关于业务流程编排器	111
关于业务流程编排器资源管理器	111
关于业务流程编排器观察器	112
业务流程编排器配置	115
数据源	116
编辑	116
测试连接	116
数据库实例	116
模式名	116
创建表	116
用户名	116
密码	117
服务器	117
提供程序	117
人员任务管理器邮件会话	117
启用电子邮件服务	117
邮件传输主机	117
邮件传输用户	117
邮件传输密码	118
业务流程编排器资源管理器 URL	118
安全性	118
管理员用户	118
管理员组	118
巡视员用户	118
巡视员组	118
JMS 认证用户	119
JMS 认证密码和确认密码	119
JMS API 认证用户	119
JMS API 认证密码和确认密码	119
升级用户认证用户	119
升级用户认证密码和确认密码	119
状态观察器	120

业务流程管理器的审计日志记录	120
人员任务管理器的审计日志记录	120
业务流程管理器的公共事件基础结构日志记录	120
人员任务管理器的公共事件基础结构日志记录	120
SCA 绑定	120
主机	120
业务流程管理器的上下文根	120
人员任务管理器的上下文根	121
相对路径	121
总线	121
使用缺省配置	121
总线成员位置	121
远程目标位置	121
新建	122
编辑	122
测试连接	122
数据库实例	122
模式名	122
创建表	122
用户名	122
密码	123
服务器	123
提供程序	123
业务流程编排器资源管理器设置	123
上下文根	123
资源管理器搜索结果限制	124
受管业务流程编排器容器	124
业务流程编排器观察器设置	124
上下文根	124
对此业务流程编排器事件收集器中的监视数据进行可视化	124
第 4 章 配置业务流程编排器	125
使用安装程序或概要文件管理工具来配置业务流程编排器	125
使用管理控制台的部署环境向导来配置业务流程编排器	127
使用管理控制台的“业务流程编排器”配置页面	129
使用 bpeconfig.jacl 脚本配置业务流程编排器	140
bpeconfig.jacl 脚本文件	146
使用生成的 SQL 脚本为业务流程编排器创建数据库模式	159
使用 SQL 脚本为业务流程编排器创建数据库	162
为业务流程编排器创建 Derby 数据库	164
为业务流程编排器创建 DB2 i5/OS 版数据库	165
为业务流程编排器创建 DB2 Linux 版、UNIX 版和 Windows 版数据库	165
为业务流程编排器创建 DB2 z/OS 版数据库	167
为业务流程编排器创建 Informix Dynamic Server 数据库	169
为业务流程编排器创建 Microsoft SQL Server 数据库	170
为业务流程编排器创建 Oracle 数据库	172
配置人员目录提供程序	173
配置虚拟成员管理器人员目录提供程序	173
配置 LDAP 人员目录提供程序	175

配置人员替代功能	180
创建业务流程编排器的队列管理器和队列	183
为业务流程编排器创建集群式队列管理器和队列	184
概述: 配置业务流程编排器资源管理器	187
配置业务流程编排器资源管理器	188
配置业务流程编排器观察器	191
除去业务流程编排器观察器 V6.0.1 样本	192
为业务流程编排器观察器准备数据库	193
在 Java 用户定义函数与 SQL 用户定义函数之间进行选择	218
配置业务流程编排器事件收集器应用程序	224
配置业务流程编排器观察器应用程序	230
对业务流程编排器启用日志记录	232
更改业务流程编排器观察器的配置参数	234
验证业务流程编排器观察器	242
激活业务流程编排器	242
验证业务流程编排器的运行是否正常	243
了解业务流程编排器的启动行为	244
联合配置了业务流程编排器的独立节点	244

第 5 章 除去业务流程编排器配置 . . . 247

使用脚本除去业务流程编排器配置	247
使用工具除去业务流程编排器观察器和事件收集器	250
使用管理控制台除去业务流程编排器配置	250
使用管理控制台除去业务流程编排器事件收集器	255
使用管理控制台除去业务流程编排器观察器	256

第 3 部分 管理 257

第 6 章 管理业务流程编排器 259

使用管理控制台管理业务流程编排器	259
管理服务器的补偿服务	259
使用管理控制台查询和重放已失败的消息	259
使用管理控制台刷新人员查询结果	261
使用管理控制台启用公共基本事件和审计跟踪	261
使用刷新守护程序刷新人员查询结果	263
使用脚本管理业务流程编排器	264
使用管理命令删除审计日志条目	264
删除无效的流程模板	266
删除无效的人员任务模板	269
删除已完成的流程实例	271
从观察器数据库中删除数据	273
使用管理命令查询和重放已失败的消息	276
使用管理命令刷新人员查询结果	279
使用管理命令除去未使用的人员查询结果	282

第 7 章 业务流程编排器资源管理器入门 285

启动业务流程编排器资源管理器	289
定制业务流程编排器资源管理器	290
为不同的用户组定制业务流程编排器资源管理器界面	290
对业务流程编排器资源管理器界面进行个性化定制	293
更改缺省 Web 应用程序的外观	293

第 8 章 业务流程编排器观察器入门 . . 299

第 9 章 管理业务流程和人员任务 303

管理流程模板和流程实例	303
使用管理控制台停止和启动流程模板	305
使用管理命令停止和启动流程模板	306
管理流程的生命周期	307
修复流程和活动	311
管理任务模板和任务实例	313
使用管理控制台停止和启动任务模板	313
使用管理命令停止和启动任务模板	314
创建和启动任务实例	315
处理任务	315
暂挂和恢复任务实例	316
管理人员任务的优先级	317
管理工作分配	317
查看任务升级	323
在业务流程编排器资源管理器中创建和编辑定制属性	325
创建业务流程报告和活动报告	325
使用预定义的列表和图表	329
创建用户定义报告	334
使用已保存的用户定义报告定义	344

第 4 部分 开发和部署模块 349

第 10 章 为业务流程和任务开发客户机应用程序 351

第 11 章 为业务流程和人员任务开发 EJB 客户机应用程序 353

访问 EJB API	354
访问会话 Bean 的远程接口	354
访问会话 Bean 的本地接口	356
查询与业务流程和任务相关的对象	358
在查询中使用变量对数据进行过滤	385
管理存储查询	386
为业务流程开发应用程序	389
对流程实例执行的操作所需的角色	389
对业务流程活动执行的操作所需的角色	390
管理业务流程的生命周期	391
处理人员任务活动	398
处理单一人员工作流程	399
向处于“正在等待”状态的活动发送消息	401
处理事件	402
分析流程结果	402
修复活动	403
BusinessFlowManagerService 接口	405
为人员任务开发应用程序	407
启动调用了同步接口的调用任务	407
启动调用了异步接口的调用任务	408
创建和启动任务实例	409
处理待执行任务或协作任务	410
暂挂和恢复任务实例	411
分析任务结果	412
终止任务实例	413
删除任务实例	413
释放已声明的任务	413

管理工作项	414
在运行时创建任务模板和任务实例	415
HumanTaskManagerService 接口	422
为业务流程和人员任务开发应用程序	424
确定可以启动的流程模板或活动	425
处理包含人员任务的单一人员工作流程	427
处理异常和故障	429
处理 API 异常	429
检查为活动设置了哪个故障	430
检查已停止的调用活动所发生的故障	430

第 12 章 开发 Web Service API 客户机应用程序 433

简介: Web Service	433
Web Service 组件与控制序列	433
Web Service API 概述	434
对业务流程和人员任务的要求	434
开发客户机应用程序	435
复制工件	435
从服务器环境中发布和导出工件	435
使用客户机 CD 上的文件	440
在 Java Web Service 环境中开发客户机应用程序	443
生成代理客户机 (Java Web Service)	443
创建 BPEL 流程的辅助控件类 (Java Web Service)	447
创建客户机应用程序 (Java Web Service)	448
添加安全性 (Java Web Service)	449
添加事务支持 (Java Web Service)	452
在 .NET 环境中开发客户机应用程序	453
生成代理客户机 (.NET)	453
创建 BPEL 流程的辅助控件类 (.NET)	454
创建客户机应用程序 (.NET)	456
添加安全性 (.NET)	457
查询与业务流程和任务相关的对象	458
对业务流程以及与任务相关的对象执行的查询	458
对业务流程对象和人员任务对象执行的查询的预定义视图	461
管理存储查询	462

第 13 章 开发 JMS 客户机应用程序 463

JMS 简介	463
业务流程的要求	463
访问 JMS 接口	464
业务流程编排器 JMS 消息的结构	465
JMS 授权的呈示	467
JMS API 概述	467
开发 JMS 应用程序	468
复制工件	469
检查响应消息以查找业务异常	469

第 14 章 使用 JSF 组件为业务流程和人员任务开发 Web 应用程序 471

在 JSF 应用程序中添加 List 组件	476
在 JSF 应用程序中添加 Details 组件	482
在 JSF 应用程序中添加 CommandBar 组件	484
在 JSF 应用程序中添加 Message 组件	487

第 15 章 开发用于显示任务消息和流程消息的 JSP 页面	491
用户定义 JSP 片段	492
第 16 章 创建插件以定制人员任务功能	493
创建 API 事件处理程序	493
API 事件处理程序	494
创建通知事件处理程序	495
创建插件以便对人员查询结果进行后处理	496
安装插件	498
注册插件	499
第 17 章 安装业务流程应用程序和人员任务应用程序	501
以交互方式安装业务流程应用程序和人员任务应用程序	503
配置流程应用程序数据源和引用设置	503
使用管理控制台卸载业务流程应用程序和人员任务应用程序	504
使用管理命令卸载业务流程应用程序和人员任务应用程序	506
第 5 部分 监视业务流程和任务	509
第 18 章 监视业务流程和人员任务	511
第 19 章 监视业务流程事件	513
业务流程所特有的事件数据	513
业务流程事件的扩展名	517
业务流程事件	525
业务流程事件中的情况	530
第 20 章 监视人员任务事件	533
人员任务所特有的事件数据	533
人员任务事件的扩展名	534
人员任务事件	537
人员任务事件中的情况	539
第 6 部分 调整	541
第 21 章 调整业务流程	543
调整长时间运行流程	544
平衡硬件资源	544
指定 DB2 数据库初始设置	545
指定 Oracle 数据库初始设置	548
规划消息传递引擎设置	549
调整应用程序服务器	549
对数据库进行微调	551
对消息传递提供程序进行微调	555
调整微流程	555
对包含人员任务的业务流程进行调整	556
减少对人员任务进行的并发访问	556
缩短查询响应时间	556
避免扫描整个表	557
优化任务查询和流程查询	557

第 22 章 调整业务流程编排器资源管理器	559
第 23 章 调整业务流程编排器观察器	561
第 7 部分 故障诊断	563
第 24 章 业务流程编排器配置故障诊断	565
业务流程编排器日志文件	565
业务流程编排器数据库和数据源故障诊断	566
启用替代功能后，任务容器无法启动	567
6.0.x 业务流程编排器 API 客户机在 6.1 环境中发生故障	568
对业务流程编排器启用跟踪	569
第 25 章 业务流程和人员任务故障诊断	571
对业务流程应用程序和人员任务应用程序的安装进行故障诊断	571
对业务流程的执行进行故障诊断	572
停止包含微流程的应用程序时发生	
ClassCastException	572
调用 processMessage 方法期间发生意外的异常（消息：CNTR0020E）	572
XPath 查询从数组中返回了意外的值	572
活动由于未处理的故障而停止（消息：CWWBE00571）	573
未对微流程进行补偿	573
长时间运行流程似乎已停止	574
调用另一个 EAR 文件中的同步子流程失败	574
执行期间发生意外的异常（消息：CWWBA0010E）	575
未知的事件（消息：CWWBE0037E）	575
找不到并且无法创建流程实例（消息：CWWBA0140E）	575
流程实例的“已失败”状态不允许执行所请求的 sendMessage 操作（消息：CWWBE0126E）	575
Java 片段包含未初始化的变量或者发生 NullPointerException	576
标准故障异常“missingReply”（消息：CWWBE0071E）	576
并行路径序列化	576
将嵌套的数据对象复制到另一个数据对象将破坏源对象中的引用	577
CScope 不可用	577
处理与流程或任务相关的消息	577
对业务流程和人员任务的管理进行故障诊断	578
升级电子邮件故障诊断	579
人员分配故障诊断	580
业务流程编排器资源管理器故障诊断	584
业务流程编排器观察器故障诊断	585
使用与流程相关的审计跟踪信息以及与任务相关的审计跟踪信息	586
业务流程的审计事件类型	587
人员任务的审计事件类型	589
业务流程的审计跟踪数据库视图结构	590

人员任务的审计跟踪数据库视图结构	594	声明	599
第 8 部分 附属资料	597		

第 1 部分 WebSphere Process Server 中的业务流程和人员任务

第 1 章 关于业务流程

业务流程是一组与业务相关的活动，您通过调用这些活动来实现业务目标。

Web Service 业务流程执行语言（WS-BPEL）中定义的流程由下列各项组成：

- 作为流程中各个步骤的活动。活动可以具有各种不同的类型。并且，活动可以分为基本活动或结构化活动。
 - 基本活动是未包含其他活动的无结构活动，例如分配活动或调用活动。
 - 结构化活动是包含其他活动的活动，例如序列活动或并行活动。
- 伙伴链接。这些链接也称为接口伙伴或引用伙伴，它们指定使用 WSDL 接口与外部伙伴进行的交互。
- 变量。这些变量存储与流程交换的数据以及在活动之间传递的数据。
- 关联集，用于使多个服务交互与同一个业务流程实例相关联。关联集基于与流程交换的消息中包含的应用程序数据。
- 故障处理程序，用于处理业务流程运行期间发生的异常情况。
- 事件处理程序，用于与正常执行的流程同步接收并处理自发消息。
- 补偿处理程序，用于指定单个活动、一组活动或作用域的补偿逻辑。

有关这些构造的更多信息，请参阅 BPEL 规范。

业务流程编排器还支持 IBM® 对 BPEL 语言的扩展，例如：

- 用于进行人员交互的人员任务活动。这些直接插入待执行任务可以是业务流程中涉及人员的步骤，例如填写表单以及核准文档等等。
- 用于运行直接插入 Java™ 代码的脚本活动。Java 代码可以访问所有 BPEL 变量、关联属性、伙伴链接以及流程上下文和活动上下文。
- 用于直接访问 WebSphere® 信息服务器或关系数据库的信息服务器活动。
- 用于进行流程模型版本控制的生效时间戳记。
- 用于以手动方式设置或控制业务流程中事务边界的扩展。
- 活动超时。

相关任务

第 303 页的第 9 章，『管理业务流程和人员任务』

业务流程和人员任务作为企业应用程序的一部分进行部署和安装。您可以使用管理控制台或管理命令来管理流程模板和任务模板，并可以使用业务流程编排器资源管理器来处理流程实例和任务实例。您可以使用业务流程编排器观察器来创建有关业务流程和人员任务的报告。

第 303 页的『管理流程模板和流程实例』

使用管理控制台或管理命令来管理流程模板。使用业务流程编排器资源管理器来处理流程实例。

第 305 页的『使用管理控制台停止和启动流程模板』

可以使用管理控制台来单独启动和停止每个已安装的流程模板。

第 306 页的『使用管理命令停止和启动流程模板』

可以使用管理命令来代替使用管理控制台停止和启动流程模板。本节阐述如何使用管理命令来停止企业应用程序中的所有流程模板。

第 307 页的『管理流程的生命周期』

流程启动后，它将进入各种状态并最终结束。作为流程管理员，可以对处于生命周期各个阶段的流程执行各种操作。

第 307 页的『启动新的流程实例』

您可以根据有权使用的流程模板来启动新的流程实例。

第 308 页的『监视流程实例的进度』

可以对流程实例的进度进行监视，以确定是否需要执行操作以使该流程能够运行到完成。

第 308 页的『暂挂和恢复流程实例』

可以将顶层长时间运行流程实例暂挂。例如，通过执行此操作，可以配置对该流程稍后要使用的后端系统的访问或者解决导致流程实例失败的问题。满足该流程的先决条件后，可以恢复该流程实例。

第 309 页的『终止流程实例』

在某些情况下，您可能想终止流程实例，例如，不再需要该流程实例所代表的工作或文档、没有任何用户能够完成该流程实例以及流程模板有问题并需要进行重新设计等等。

第 310 页的『删除流程实例』

可以将流程模板设计成在流程实例完成时不自动将它们删除。在这些流程实例完成后，您可以手动将其删除。

第 311 页的『修复流程和活动』

如果流程遇到问题，您可以对其进行分析，然后修复活动。

第 311 页的『重新启动活动』

在修复活动之后，可以使用新的输入数据来将其重新启动。

第 312 页的『强制完成活动』


如果您知道活动无法及时完成，例如由于所调用的服务不再可用而无法及时完成，那么可以强制完成该活动以使流程可以继续运行。

第 312 页的『管理微流程的补偿』

微流程运行期间，它可能会遇到问题。对于这些情况，在流程模型中可能为流程定义了补偿。补偿允许您撤销先前已完成的步骤，例如，以便复位数据和状态，从而能够从这些问题中恢复。

相关信息

 [Web Service 业务流程执行语言 V1.1](#)

 [OASIS Web Service 业务流程执行语言 V2.0](#)

流程模板

流程模板是部署并安装在运行时环境中的流程定义。

除了定义流程时指定的属性以外，已安装的业务流程还可以处于下列其中一种状态：

已启动 创建并启动流程模板后，就可以启动该模板的新实例。

已停止 在可以卸载业务流程应用程序之前，必须停止流程模板。流程模板进入“已停止”状态后，就无法创建和启动此模板的新实例。

业务流程类型

业务流程可以是长时间运行流程或微流程。

长时间运行流程

长时间运行业务流程是可中断的流程，并且它的每个步骤都可以在自己的物理事务中运行。长时间运行业务流程可以等待外部激励。外部激励的示例包括，由企业到企业交互中另一业务流程发送的事件、对异步调用的响应或者人员任务完成。

长时间运行流程具有下列特征：

- 在多个事务中运行。
- 以同步方式或异步方式与服务进行交互。
- 它的状态存储在运行时数据库中，这使流程支持正向恢复

微流程

微流程从头到尾在一个物理线程中运行，而不会中断。微流程有时被称为不可中断的业务流程。微流程可以具有不同的事务功能。微流程参与工作单元，后者可以是全局事务或活动会话。

微流程具有下列特征：

- 在一个事务或活动会话中运行
- 运行时间通常较短
- 它的状态是瞬态的，因此不会存储在运行时数据库中
- 它通常以同步方式调用服务
- 它只能有不可中断的子流程
- 它不能包含：
 - 人员任务
 - 等待活动
 - 非始发接收或检取活动

相关概念

第 15 页的『业务流程交互的影响因素』

许多因素将对各种调用方案中业务流程的行为产生影响。这些因素包括交互样式、业务流程类型、操作类型和服务端点解析。

流程版本控制

可以创建新版本的业务流程，以使同一流程的多个版本在运行时环境中共存。

在 WebSphere Integration Developer 中定义业务流程时，可以指定版本控制信息，例如生效日期。流程的版本由其生效日期确定。这意味着，不同版本的流程可以具有相同的流程名称，但它们的生效日期不同。在运行时使用的流程版本由该流程是用于早期绑定方案还是后期绑定方案确定。

早期绑定

在早期绑定方案中，将在建模期间或者部署流程时确定所调用的流程版本。调

用者将调用以静态方式绑定的专用流程。即使存在另一个根据不同版本的生效日期确定有效的流程版本，也将调用当前以静态方式连接的流程并忽略所有其他版本。

SCA 连接是早期绑定的一个示例。如果将独立引用连接到某个流程组件，那么使用此引用进行的每次流程调用都将针对该流程组件所代表的特定版本进行。

后期绑定

在后期绑定方案中，调用者在调用流程时确定所要调用的流程模板。在这种情况下，将使用当前有效的流程版本。当前有效的流程版本将取代所有先前版本的流程。现有流程实例将继续使用启动时与其相关联的流程模板运行。因此，流程模板分为下列三类：

- 用于新流程实例的当前有效流程模板
- 仍用于现有长时间运行流程实例但已不再有效的流程模板
- 根据生效日期，未来将会生效的流程模板

要在调用子流程时应用后期绑定，父流程必须指定子流程模板的名称，在引用伙伴中将根据这个子流程模板选择有效的子流程。将使用流程的生效时间属性来确定当前有效的子流程模板。

在业务流程编排器资源管理器中调用新流程是后期绑定的一个示例。创建的实例始终基于具有过去生效日期的当前有效流程版本。

流程实例

流程实例是流程模板的有状态表现形式。

使用 Web Service 业务流程执行语言（WS-BPEL）定义的业务流程代表有状态的 Web Service，因此，它们可以与其他 Web Service 进行长时间交互。每当 BPEL 流程启动时，将创建该流程的新实例，该实例能够与其他业务合作伙伴进行通信。实例在它的最后一个活动完成时、终止活动运行时或者该实例遇到流程未处理的故障时完成。

流程生命周期

流程启动时，业务流程实例的处理即开始，并且它将开始与所在环境进行交互。这意味着，某些交互只有在特定流程状态下才有可能进行，这些交互又将影响该流程实例的状态。

流程实例的状态过渡图

在流程实例的生命周期内，每当发生重要的情况时，流程将更改状态。例如，某个 API 请求使处于“正在运行”状态的流程进入“已暂挂”状态。状态过渡图说明流程生命周期内可能发生的状态过渡。

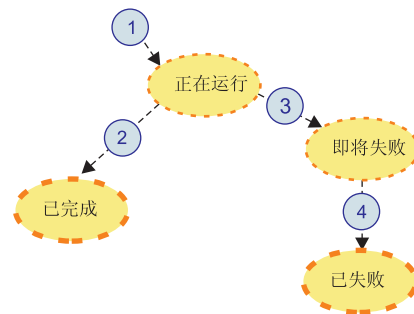
这些图中使用的约定

图中的状态过渡由编号指示。随后的支持文本对这些编号进行说明。此外，这些图包含下列类型的符号：

符号	说明
	瞬态状态。这些状态不可视。
	持久状态。
	瞬态结束状态。
	持久结束状态。
	由业务流程管理器自动触发的状态过渡。
	由使用 API 进行的外部交互引起的状态过渡。
	由业务流程管理器控制的状态过渡，或者由使用 API 进行的外部交互引起的状态过渡。

微流程实例的状态过渡图

对于微流程而言，由于流程始终在一个事务中运行，并且不会将实例信息保存下来以便用于流程实例导航，因此微流程被认为是无状态流程。但是，根据流程定义以及业务流程管理器配置方式的不同，可以在公共基本事件或审计日志中公布微流程的状态。



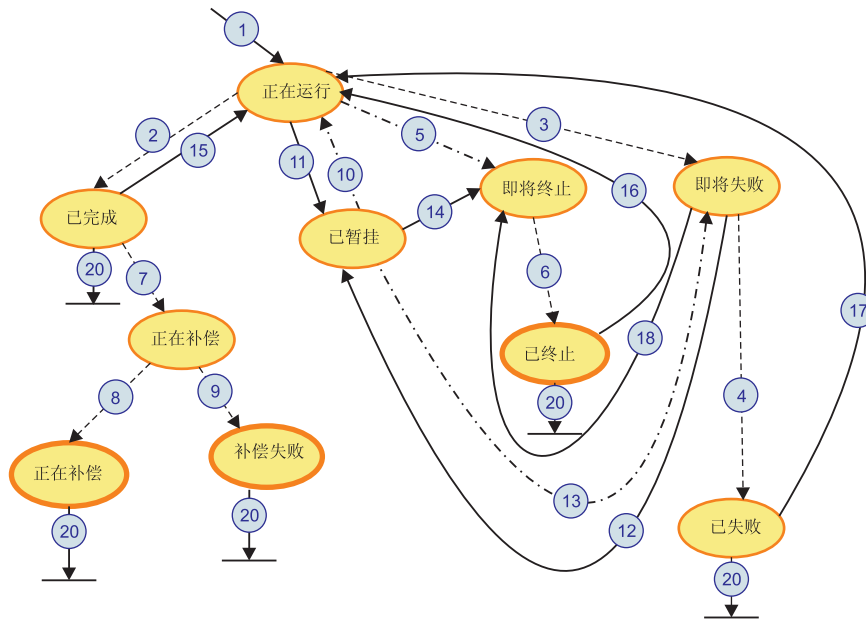
下图说明微流程实例可以具有的状态。

流程实例正常启动后，它进入的第一个流程状态是“正在运行”状态（1）。流程实例正常运行完成时，流程状态将从“正在运行”更改为“已完成”（2）。如果某个故障到达流程边界，那么该流程将进入“即将失败”状态（3）。在故障处理程序运行期间，该流程将一直处于“即将失败”状态。接着，该流程实例将进入“已失败”状态（4）。

所有这些状态过渡都由业务流程管理器触发。微流程启动后，您无法对这些自动步骤施加影响。

长时间运行流程实例的状态过渡图

长时间运行流程在多个事务中运行。长时间运行流程的状态具有持久性，因此可视。下图说明长时间运行流程实例的生命周期内可能出现的状态过渡。



“正在运行”、“已完成”、“即将失败”和“已失败”状态以及它们之间的状态过渡与微流程相同。

流程实例由外部请求或终止活动终止。流程实例的终止可以跨多个导航步骤并因此跨多个链接的事务，例如，在终止长时间运行活动或子流程时，情况即如此。在这个终止阶段，流程实例处于“即将终止”状态（5）（14）（18）。流程的所有长时间运行部件终止后，流程实例的状态也将更改为“已终止”（6）。

如果子流程成功结束，但父流程随后失败，那么可以对该子流程进行补偿。在补偿期间，该子流程将处于“正在补偿”状态（7）。如果成功完成补偿，那么该子流程将进入“已补偿”状态（8）。如果补偿不成功，那么该子流程将进入“补偿失败”状态（9）。这些状态事务由父流程自动启动。

如果流程实例的导航仍处于活动状态，即，该流程实例处于“正在运行”或“即将失败”状态，那么可以通过 API 请求将其暂挂。接着，它将在达到指定时间后被重新激活，或者由恢复请求重新激活。对于暂挂请求而言，流程状态将从“正在运行”或“即将失败”更改为“已暂挂”（11）（12）；对于恢复请求而言，流程状态将从“已暂挂”更改为“正在运行”或“即将失败”（10）（13）。此外，还可以将处于“已暂挂”状态的流程终止（14）。只能暂挂和恢复顶层流程实例。但是，暂挂或恢复状态将传播到子流程。

流程进入其中一种结束状态（即“已完成”、“已终止”或“已失败”状态）之后，可以通过重新启动 API 请求将其重新启动（15）（16）（17）。只能重新启动顶层流程实例，并且只能对子流程实例进行补偿。

流程实例进入结束状态后，可以将其删除（20）。如果适当地设置完成时自动删除属性，那么可以自动删除该流程，此外，也可以通过明确的删除请求触发删除。

子流程的生命周期管理

由另一个流程启动的流程称为子流程。子流程的生命周期管理方式取决于这些流程的设计。

通常，为了实现模块化以及进行复用，最好将业务逻辑的一个或多个步骤作为独立的流程实现并从主流程中调用此流程。子流程还可以启动另一个流程。这样，可以产生流程实例层次结构。部署这些流程时，必须将“流程与流程”关系中的所有流程模板部署到同一个业务流程编排器数据库中。

子流程与主调流程之间的关系可以是对等关系或父子关系。此关系决定了对主调流程调用流程生命周期管理操作时子流程的行为。生命周期操作包括暂挂、恢复、终止、删除和补偿。在父子关系中，只能对顶层流程实例执行流程生命周期管理操作。

流程与子流程的关系由子流程的 `autonomy` 属性确定。此属性可以具有下列其中一个值：

Peer 对等流程被视为顶层流程。顶层流程是一个流程实例，它未被任何其他流程实例调用或者仅被 `autonomy` 属性值为 `peer` 的流程实例调用。如果子流程包含在对等关系中，那么对主调流程实例执行的生命周期操作不会传播到子流程实例。

使用单向接口创建和启动的长时间运行流程被视为对等流程。在运行时，它的 `autonomy` 属性将被忽略。

Child 如果子流程包含在父子关系中，那么对父流程实例执行的生命周期操作将应用于子流程实例。例如，如果将父流程实例暂挂，那么还会将 `autonomy` 属性值为 `child` 的所有子流程实例暂挂。子流程只有在完成后才能返回到它的父流程，即，子流程的最后一个操作必须是对主调父流程的应答。请确保流程逻辑中所有可能的路径结束时，都将应答活动作为该路径中的最后一个操作。

微流程始终作为子流程运行，即，它的 `autonomy` 将被忽略。

只能在直接进行交互的流程之间建立父子关系。如果另一个 SCA 组件拦截了此交互，例如在两个流程组件之间连接了接口映射组件，那么可能会导致无法建立父子关系。

活动的状态过渡图

在活动实例的执行过程中，当遇到重要步骤时，该活动实例的状态将更改。状态和状态过渡随活动类型的不同而有所变化。

在基本活动的生命周期中，状态和状态过渡十分重要。基本活动分为下列活动类型。状态过渡图随活动类型的不同而有所变化：

- 短时间运行活动，例如分配活动、清空活动、应答活动、重新抛出活动、抛出活动和终止活动
- 等待外部事件的活动，例如接收活动和等待活动
- 检取（接收选项）活动
- Java 片段活动
- 调用活动

- 人员任务活动

与流程实例的状态图不同，活动结束状态不明确公开。活动的生命周期取决于外层流程。活动始终与流程实例一起被删除。

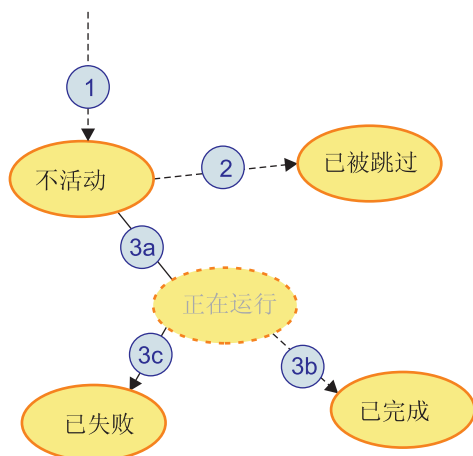
这些图中使用的约定

图中的状态过渡由编号指示。随后的支持文本对这些编号进行说明。此外，这些图包含下列类型的符号：

符号	说明
	瞬态状态。这些状态不可视。
	持久状态。
	由业务流程管理器自动触发的状态过渡。
	用户交互引起的状态过渡，例如由 API 请求引起的状态过渡。
	由业务流程管理器或用户交互控制的状态过渡。

短时间运行活动类型的状态过渡图

以下状态图显示简单短时间运行活动类型的状态和状态过渡，例如：分配活动、清空活动、应答活动、重新抛出活动、抛出活动和终止活动。此图介绍了“不活动”、“已被跳过”、“已完成”和“已失败”状态。这些状态是所有基本活动类型的公共状态。



创建活动后，它处于“不活动”状态（1）。封闭在流中的活动可以有多个入局链接和连接条件。在可以启动此类活动前，必须对所有入局链接进行导航。活动的 **suppressJoinFailure** 属性和连接条件求值结果确定了该活动的后续行为：

- 连接条件求值为 false 并且 **suppressJoinFailure** 属性设置为 true。

活动状态更改为“已被跳过”（2），并且离开该活动的链接将作为死路径进行导航。

- 连接条件求值为 false 并且 **suppressJoinFailure** 属性设置为 false。

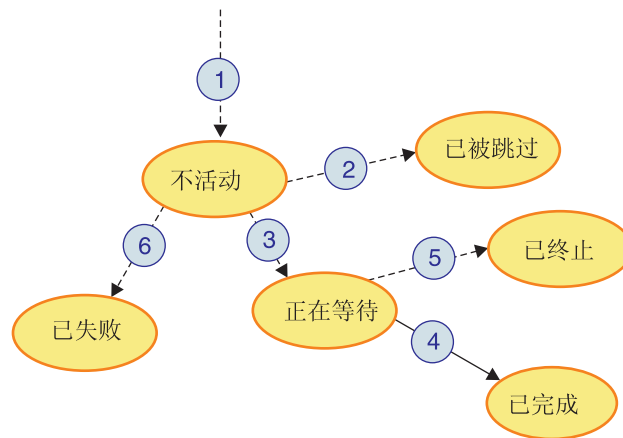
该活动尚未启动，因此处于“不活动”状态，并且将发生 `bpmw:joinFailure` 标准故障。

- 连接条件求值为 true。

对于未封闭在流中的活动而言，这是预期内的行为。该活动将被激活，并且它的状态将更改为“正在运行”（3a）。将运行该活动实现，接着活动状态更改为“已完成”（3b）。如果该实现失败，例如分配活动中的复制语句语法不正确，那么该活动的状态将更改为“已失败”（3c）。所有短时间运行活动都不可中断。因此，“正在运行”状态永远不可视。

等待外部事件的活动的状态过渡图

下图说明等待或接收活动的生命周期内可能出现的状态和状态过渡。



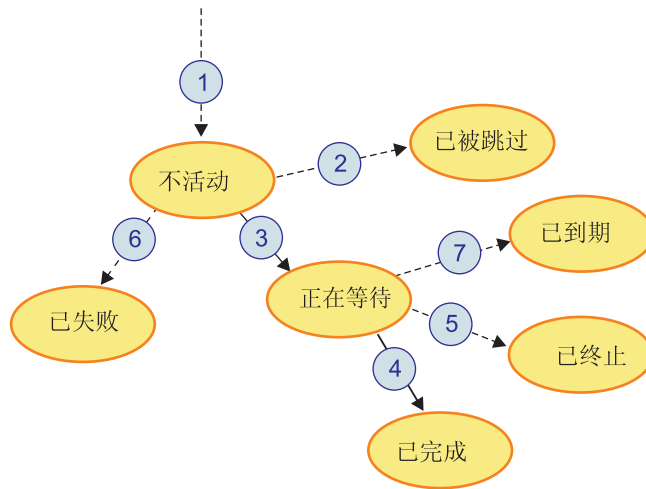
接收活动和等待活动的开始阶段与短时间运行活动相同。但是，在接收活动和等待活动被激活后，状态将更改为“正在等待”而不是“正在运行”（3）。现在，接收活动或等待活动已准备好接收外部请求或根据指定的超时时间进行等待，然后才能完成并进入“已完成”状态（4）。对于接收活动而言，到“已完成”状态的过渡由所接收的消息触发。对于等待活动而言，此过渡在达到所指定等待时间后自动进行。

等待活动或接收活动的实现比简单的短时间运行活动类型复杂。等待活动或接收活动可能会在活动开始完成前失败，例如，对等待活动进行的等待时间求值操作失败。此故障将导致该活动在进入“正在等待”状态前进入“已失败”状态（6）。

当活动处于“正在等待”状态时，外层流程可能会接收到终止请求，与等待活动或接收活动并行的分支也可能发生故障。如果发生任何这些事件，那么等待活动或接收活动将终止，并且活动状态将更改为“已终止”（5）。

检取（接收选项）活动的状态过渡图

以下状态图显示检取活动（也称为接收选项活动）的状态和状态过渡。

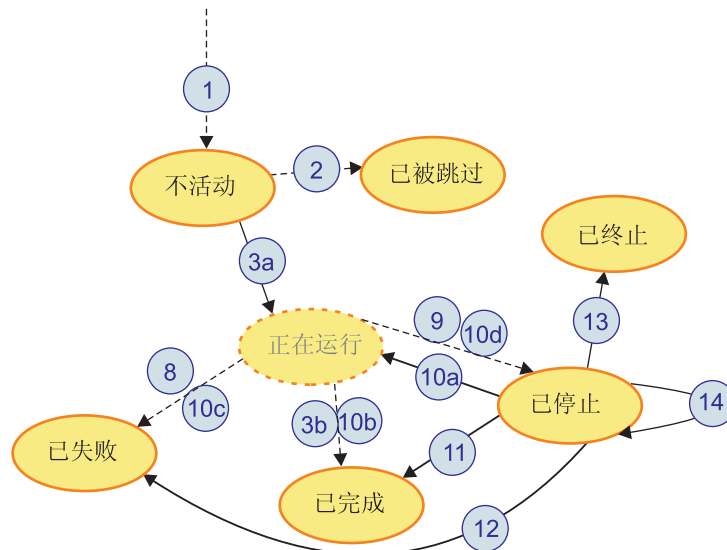


检取活动的状态和状态过渡（1）至（6）与接收活动相同。

此外，如果处于等待状态的检取活动的警报分支在检取活动请求到达前被激活，那么检取活动将到期。于是，该活动将进入“已到期”状态（7）。

Java 片段活动的状态过渡图

下图说明 Java 片段活动的生命周内可能出现的状态和状态过渡。



Java 片段活动的运行时行为与短时间运行活动的运行时行为类似；当 Java 片段活动被激活时（3a），将运行实现。因为 Java 片段的实现较短，所以“正在运行”状态不可视。如果 Java 片段成功完成，那么活动状态将立即更改为“已完成”（3b）。

如果 Java 片段失败，那么活动可能会采用下列其中一种行为：

- 活动状态更改为“已停止”（9）。如果该活动的 **continueOnError** 属性设置为 false，并且 Java 片段的外层作用域内的故障处理程序无法捕获该 Java 片段抛出的故障，那么需要进行管理性交互，以使流程导航可以继续进行。
- 活动状态更改为“已失败”（8），故障处理开始。

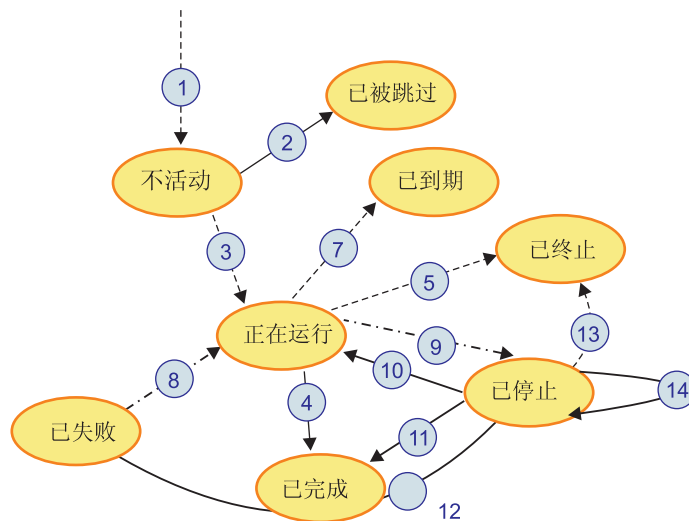
如果活动处于“已停止”状态，那么可以使用 API 请求对其进行强制重试或强制完成。如果对其进行强制重试，那么该活动将进入“正在运行”状态，并且将再次运行实现（10a）。因为外部或流程内部情况已更改，所以 Java 片段可以成功完成，并且活动状态将更改为“已完成”（10b）。如果该活动再次失败，那么管理员可以使用强制重试 API 的实现来停止该活动（10d），也可以让流程故障处理功能处理故障（10c）。

当该活动处于“已停止”状态时，如果外层流程或作用域终止，那么该活动状态将更改为“已终止”（13）。

如果该活动被强制完成，那么其实现将不再运行。该活动将进入“已完成”（11）、“已失败”（12）或“已停止”状态（14）。

调用活动的状态过渡图

下图说明采用异步实现方式的调用活动的生命周期内可能出现的状态和状态过渡。如果服务应答在服务请求事务的后续事务内发生，那么该实现就是异步实现。流程和事务组件的 SCA 限定符确定是以同步方式还是异步方式调用服务。



调用活动的激活与所有其他活动类型的激活相同（1）（2）。

当调用活动正常运行至完成时，该活动将启动，并且状态将更改为“正在运行”（3）。如果服务调用成功返回，那么该活动将进入“已完成”状态（4）。

只要服务未接收到应答并且活动处于“已停止”状态，管理员就可以强制重试或强制完成该活动。如果无法应答服务，例如由于系统运行中断而无法应答服务，那么此功能很实用。相应的 API 也可能使状态从“正在运行”过渡到“已停止”（9）、“已失败”（8）和“已完成”（4）。如果该异步服务是子流程，那么当该活动处于“正在运行”状态时，不能强制重试或强制完成该活动。

与 Java 片段活动相同，调用活动可以停止（9）。接着，它可以由管理操作修复，也可以因为外层作用域或流程终止而终止（13）。

如果对该活动定义了到期时间，那么处于“正在运行”状态的活动将到期。于是，该活动将进入“已到期”状态（7）并抛出超时故障。故障处理程序可以处理此故障。

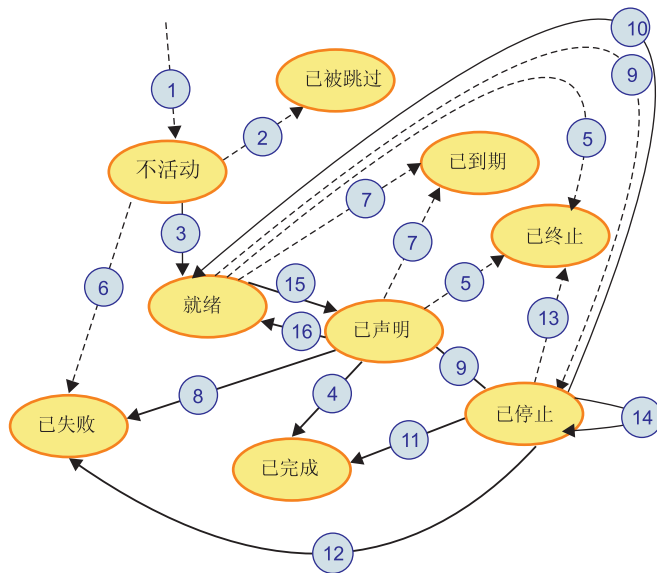
例如，如果该活动的外层作用域由于流程中某条并行路径发生故障而终止，并且该活动处于“正在运行”状态，那么该活动也将终止并进入“已终止”状态（5）。

具有同步服务调用的调用活动的状态过渡与 Java 片段的状态过渡相同。同步调用与异步调用之间的状态和状态过渡差别如下所示：

- 具有同步服务调用的调用活动的“正在运行”状态永远不可视。
- 到期时间不适用于具有同步调用的调用活动；此活动可能永远不会进入“已到期”状态。
- 具有同步服务调用的调用活动永远不会终止。

人员任务活动的状态过渡图

下图说明人员任务活动的生命周期内可能出现的状态和状态过渡。



人员任务活动的运行时行为与调用活动类似。调用活动的“正在运行”状态与人员任务活动的“就绪”状态和“已声明”状态相对应。“就绪”状态表明该活动可供人员处理。当某位人员声明该活动以便对其进行处理时，该活动将进入“已声明”状态（15）。

处理活动的人员负责提供所需的信息并完成该活动。接着，该活动将进入“已完成”、“已失败”或“已停止”状态。此外，声明该活动的人员可能会确定该活动无法完成。于是，该人员释放该活动，以允许其他用户对其进行处理。在这种情况下，该活动将回到“就绪”状态（16）。

其他状态过渡与具有异步服务调用的调用活动相同。

相关概念

第 22 页的『业务流程中的故障处理和补偿处理』

故障是任何可能导致正常业务流程处理发生变化的异常情况。设计精良的流程应该对故障加以考虑并尽可能对其进行处理。补偿是一种故障处理方式。

业务流程的调用方案

业务流程是 SCA（服务组件体系结构）组件实现类型。它可以发布服务供其他伙伴使用，也可以使用由其他伙伴提供的服务。业务流程可以通过业务流程编排器 API 提供的服务提供程序、其他 SCA 服务组件的 SCA 服务提供程序或者调用了其他 SCA 服务组件（包括其他业务流程）的 SCA 客户机。

业务流程作为通过业务流程编排器 API 提供的服务提供程序

您可以使用业务流程管理器 API 将业务流程实例化。业务流程编排器客户机应用程序也使用此 API 将业务流程作为服务提供程序加以利用。这些客户机应用程序可以创建和启动业务流程实例，并可以查询和处理现有的流程实例。业务流程管理器 API 作为 EJB、Web Service 和 JMS 消息接口提供，可用于设计 EJB、Web Service 和 JMS 客户机。

业务流程作为其他 SCA 服务组件的 SCA 服务提供程序

在此调用方案中，业务流程代表可以由其他充当客户机的 SCA 组件调用的 SCA 组件。由于业务流程作为 SCA 组件的实现，因此您可以从 SCA 客户机和其他 SCA 组件中调用它提供的服务。这些机制包括：

- 连线，用于将 SCA 客户机（引用）连接到代表业务流程的组件的接口
- 组件引用和接口的 SCA 限定符设置，用于确定交互样式、事务行为和交互可靠性等方面的特性

业务流程作为调用其他 SCA 服务组件的 SCA 客户机

另一方面，由于业务流程是 SCA 组件的实现，因此它使用的服务可以由 SCA 组件或 SCA 导入提供。SCA 引用代表出站交互中使用的 BPEL 伙伴链接。此引用可以连接到其他 SCA 组件或导入，并可以使用 SCA 限定符使服务质量属性与交互相关联。

业务流程作为调用其他业务流程的 SCA 客户机

如果业务流程既代表 SCA 客户机也代表 SCA 服务，那么您可以在 SCA 级和业务流程级选择这两者。在 SCA 级，可以使用 SCA 连线将 SCA 客户机连接到 SCA 服务。在业务流程级，可以使伙伴链接与充当服务提供程序的业务流程的名称相关联。

业务流程交互的影响因素

许多因素将对各种调用方案中业务流程的行为产生影响。这些因素包括交互样式、业务流程类型、操作类型和服务端点解析。

交互样式

可以采用同步方式或异步方式来调用业务流程提供的操作。

要点：同步交互的合理响应时间不应超过几秒钟。如果业务流程实现的请求/响应操作未在短时间内返回结果，那么您应考虑采用同步交互样式以提高性能。以同步方式调用此类操作将导致资源分块。并且，还有可能引起超时情况，这取决于系统工作负载，因此无法事先确定。

业务流程类型

业务流程可以是微流程或长时间运行流程。每种流程类型的特征都会对调用方案产生影响。

WSDL 操作类型

SCA 引用和 SCA 接口与包含一个或多个操作的 WSDL 端口类型相关联。每个操作都可以是单向操作或请求/响应操作。单向操作是指主调客户机不可知其是否已完成的服务执行。此服务执行以成功调用相关服务作为结束。请求/响应操作是指主调客户机可知其是否已完成的服务执行。当主调客户机获得服务执行结果时，此服务执行结束。

服务端点解析

在业务流程的上下文中，调用客户机可以通过下列方式与所要调用的服务相关联：

- SCA 连线以静态方式使 SCA 引用与所调用服务的接口相关联。这是 SCA 级机制，如果客户端和/或服务端作为业务流程实现，那么可以应用此机制。
- 可以在充当 SCA 客户机的业务流程所含伙伴链接的上下文中设置端点引用。端点引用唯一地确定所要调用的 Web Service 的通信端点。通常，可以调用任何 Web Service。端点引用可以应用于 Web Service 绑定或 SCA 绑定。
- 可以对充当 SCA 客户机的业务流程所含伙伴链接设置业务流程模板名。模板名唯一地确定同一服务器或集群中部署的另一个业务流程的名称。

相关概念

第 5 页的『业务流程类型』

业务流程可以是长时间运行流程或微流程。

在业务流程与服务之间传递参数

业务流程可以使用服务组件体系结构（SCA）服务，它也可以被其他 SCA 服务使用。在 SCA 服务与流程之间交换数据的方式取决于流程的设计。

业务流程使用服务

业务流程中对服务的使用通过流程模型中的业务流程执行语言（BPEL）的调用活动实现。传递给 SCA 服务的数据是从一个或多个 BPEL 变量检索的。通常，此数据通过值进行传递，这意味着被调用服务对该数据的副本进行处理。

在某些情况下，可以通过引用来传递数据。通过引用传递数据有助于提高业务流程的性能。

如果符合下列所有条件，那么通过引用将数据传递给业务流程：

- 服务调用以同步方式进行。
- BPEL 流程与被调用服务在同一个模块中。

- 使用具有数据类型的变量来交换数据。

如果被调用服务修改了数据，这些更改将应用于相应的 BPEL 变量。但是，由于被调用服务对数据所作的任何更改都不具有持久性，因此该服务最好不要更新该数据。对于长时间运行流程而言，将在落实当前事务时废弃更改；对于微流程而言，将在该流程结束时废弃更改。此外，被调用服务更新变量时，不会生成事件。

业务流程被服务使用

其他服务使用的业务流程包含流程模型中的接收活动、检取活动或事件处理程序。传递到该流程的数据被写入一个或多个 BPEL 变量。通常，此数据通过值进行传递。

但是，如果符合下列所有条件，那么将通过引用传递数据：

- 业务流程调用以同步方式进行。
- 该服务与被调用业务流程在同一个模块中。
- 使用具有数据类型的变量来交换数据。

如果被调用流程修改了 BPEL 变量，那么还将修改主调服务中的输入数据。

业务流程的事务行为

业务流程作为事务的组成部分执行。对于长时间运行流程而言，业务流程的导航可以跨多个事务；对于微流程而言，此导航作为一个事务的组成部分进行。这样的导航事务可以由外部请求、内部消息或者来自异步服务的响应触发。事务启动时，将根据流程定义执行必需的活动。调用的服务可以参与该事务。

相关概念

第 15 页的『业务流程的调用方案』

业务流程是 SCA（服务组件体系结构）组件实现类型。它可以发布服务供其他伙伴使用，也可以使用由其他伙伴提供的服务。业务流程可以通过业务流程编排器 API 提供的服务提供程序、其他 SCA 服务组件的 SCA 服务提供程序或者调用了其他 SCA 服务组件（包括其他业务流程）的 SCA 客户机。

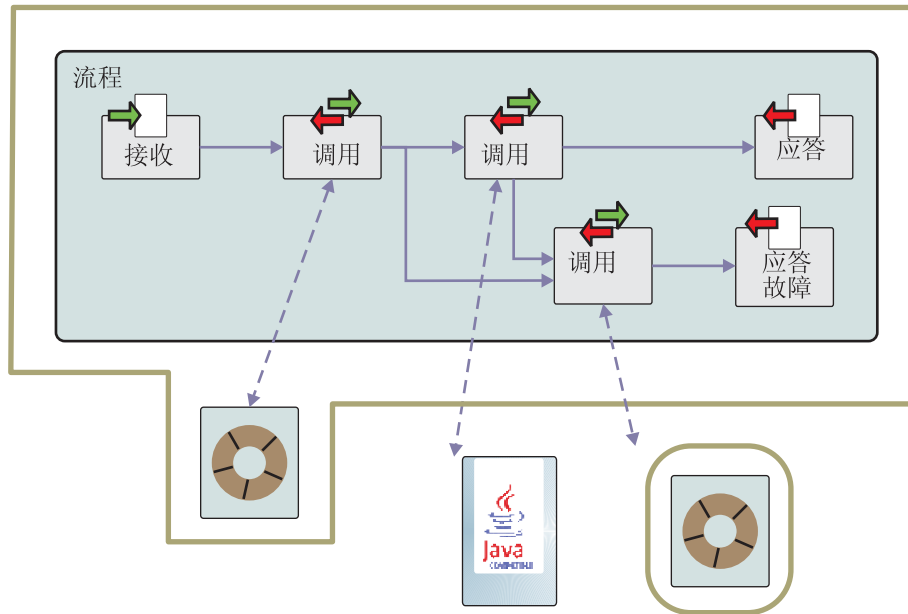
微流程的事务行为

微流程是短时间运行的流程。它们可以在事务中运行，也可以在对该微流程的 SCA 组件指定的活动会话中运行。以下内容对作为事务一部分执行的微流程进行说明。

微流程不可中断。因此，微流程不能包含等待外部事件或用户交互的活动，例如人员任务活动。

微流程是瞬态流程。微流程的流程实例状态保存在内存中，而不是存储在运行时数据库中。但是，可以将微流程实例的状态保存在审计日志或公共基本事件中。

下图说明微流程的事务以及与微流程进行交互的服务。事务边界内的服务参与微流程事务；边界外的服务不参与该事务。



调用的服务以及微流程事务

虽然微流程在一个事务中运行，但微流程的执行可以涉及多个事务。这是因为，通过调用活动调用的服务既可以参与微流程的事务也可以在它自己的事务中运行。

下列设置确定该服务是参与微流程的事务还是在它自己的事务中运行。

- 用于调用该服务的交互样式。

交互样式可以是同步调用或异步调用。此样式由目标 SCA 组件或 SCA 导入的首选交互样式以及该操作是单向操作还是请求/响应操作确定，如下表所示：

表 1.

目标组件或导入的首选交互样式	单向操作	请求/响应操作
任何	同步调用	同步调用
同步	同步调用	同步调用
异步	异步调用	同步调用

- 对该流程以及所调用服务指定的服务组件体系结构（SCA）事务限定符：
 - 对流程组件的引用指定的 **suspendTransaction** 限定符确定，该流程的事务上下文是否传播到所要调用的服务。
 - 对服务接口指定的 **joinTransaction** 限定符确定，如果传播事务的话，服务是否参与其调用者的事务。

根据这些设置的不同，下列规则适用于所调用的服务：

同步调用

joinTransaction	suspendTransaction = true	suspendTransaction = false
joinTransaction = true	此服务不参与微流程事务	此服务参与微流程事务
joinTransaction = false	此服务不参与微流程事务	此服务不参与微流程事务

如果某个服务参与微流程事务，那么仅当落实微流程事务时，才会保存该服务对事务资源所作的更改。如果某个服务未参与微流程事务，那么可以保存该服务对事务资源所作的更改，即使该事务回滚亦如此。可以使用补偿来撤销该服务所作的更改。

异步调用

服务始终在自己的事务中运行。要确保异步 SCA 消息的发送参与当前导航事务，必须将微流程的 **asynchronousInvocation** 限定符设置为 **commit**。

相关概念

第 25 页的『业务流程中的补偿』

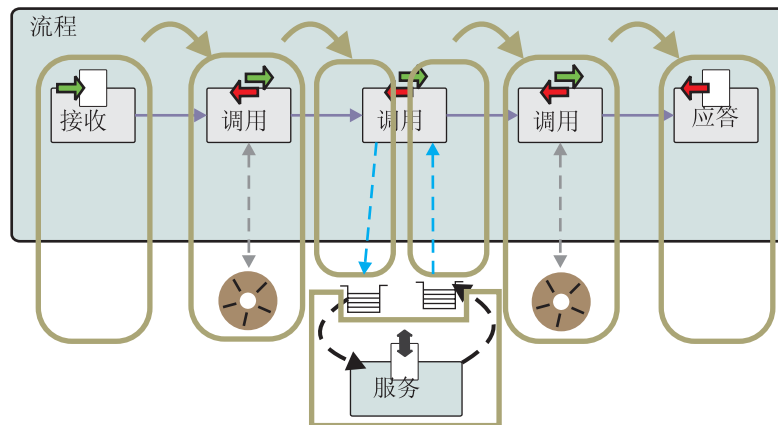
补偿用于撤销流程中已成功完成的操作。

长时间运行流程的事务行为

长时间运行流程跨多个事务。每个事务都由 Java 消息传递服务 (JMS) 消息或外部事件触发。事务可以通过将新消息放入消息队列来触发后续事务。

为了跨事务边界进行导航，流程实例及其活动实例的状态保存在数据库中。

下图说明长时间运行流程中的每个导航步骤如何在它自己的事务中执行。一个导航步骤可以跨多个活动，如调用了服务的调用活动所示。并且，在一个事务中可以运行多个活动。



以下内容对长时间运行流程的事务边界进行描述。您可以使用事务行为属性来更改事务边界。但是，业务流程管理器可以在任何时候添加或删除事务边界。

通常，在下列情况下需要事务边界：

- 在等待外部请求时，即，在流程导航期间到达接收活动或检取活动（也称为接收选项活动），但尚未接收到该活动的相应请求时
- 在为等待活动安排计时器时
- 在使用调用活动以异步方式调用服务时
- 在调用人员任务活动时

此外，在下列情况下，业务流程管理器将引入事务边界。但是，流程设计不能依赖于这些边界，这是因为，它们在流程导航期间可能会被覆盖，并且将来也可能会更改。

- 在接收到对启动该流程的接收活动或检取活动的请求之后
- 在流程导航期间发生故障时
- 在以同步方式调用未参与流程事务的服务的调用活动启动前后
- 在将生命周期操作传播到子流程时，例如，父流程暂挂后，它的子流程也将在后续事务中暂挂
- 在流程完成后要自动删除流程实例时
- 在尝试从导致跨一系列活动的事务回滚的故障进行恢复时
- 在使用事务行为属性指定要引入事务边界时

如果有保证的事务边界，那么最好将需要在单一事务中执行的业务逻辑放入一个微流程（即子流程）。微流程的逻辑始终在单一事务中运行。

修改事务边界

设计业务流程时，通过更改相应活动的事务行为属性，可以提供调用、片段和人员任务活动的建议事务边界。此属性可以具有下列其中一个值：

在此之前落实

落实当前事务，然后启动新事务。具有此属性值的活动将成为新事务的第一个活动。

在此之后落实

此活动参与当前事务。在此活动成功完成后，落实事务，然后启动新事务。对于紧跟其后的每个活动都将启动一个新事务，因此每个后续活动都将成为其中一个新事务中的第一个活动。

参与

此活动参与当前事务。不会在此活动前后设置其他事务边界。

在下列情况下，此设置允许事务根据后续活动的事务行为属性设置值继续对这些活动进行导航。

- 如果调用活动以同步方式调用服务，那么响应消息到达时，将触发新事务。该事务非常短，它将在调用活动的状态更新后立即落实。
- 在人员任务活动序列中，每个人员任务活动需要两个事务，其中一个事务用于激活该人员任务活动，另一个事务用于完成该人员任务活动。如果将此设置更改为参与，那么可以将事务数减少为每个人员任务活动一个事务。这是因为，先前人员任务活动的完成和后续活动的激活在同一个事务中执行。
- 要启用由服务器控制并使用了 `completeAndClaimSuccessor` API 的页面流。

需要自己的事务

此活动在它自己的事务中运行。这意味着，此活动启动前将落实当前事务，此活动完成后将启动新事务。

如果调用活动调用了未参与当前事务的同步服务，那么将忽略事务行为属性。对于这种情况，在调用活动启动前以及调用活动完成后，始终存在事务边界。

对流程活动中的并行分支进行并发导航

要对流程活动中的并行分支进行并发导航，在每个分支开头都需要新的事务边界，以便在独立的事务中处理每个并行活动。这意味着，必须将每个并行分支中第一个活动的事务行为属性设置为在此之前落实或需要自己的事务，以便从此流程的开头实现并行性。

注：对于 Informix®、Oracle 和 Derby 数据库系统而言，将对流程实例中并行分支的导航事务进行序列化，即，它们不能同时运行。这是因为，数据库实体锁与 DB2® 数据库锁的粒度不同。但是，此类并行分支以异步方式触发的服务仍以并行方式运行；对于这些数据库系统而言，仅对流程导航进行序列化。

长时间运行流程中的被调用服务和事务

在长时间运行流程中使用调用活动调用的服务既可以参与该长时间运行流程的当前事务，也可以在它自己的事务中运行。

下列设置确定该服务是参与长时间运行流程的事务还是在它自己的事务中运行。

- 用于调用该服务的交互样式。

交互样式可以是同步调用或异步调用。此样式由目标 SCA 组件或 SCA 导入的首选交互样式以及该操作是单向操作还是请求/响应操作确定，如下表所示：

表 2.

目标组件或导入的首选交互样式	单向操作	请求/响应操作
任何	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

- 对该流程以及所调用服务指定的服务组件体系结构（SCA）事务限定符：
 - 对流程组件的引用指定的 **suspendTransaction** 限定符确定，该流程的事务上下文是否传播到所要调用的服务。
 - 对服务接口指定的 **joinTransaction** 限定符确定，如果传播事务的话，服务是否参与其调用者的事务。

根据所设置交互样式和 SCA 限定符的不同，下列规则适用于所调用的服务：

同步调用

joinTransaction	suspendTransaction = true	suspendTransaction = false
joinTransaction = true	此服务不参与长时间运行流程的事务	此服务参与长时间运行流程的事务
joinTransaction = false	此服务不参与长时间运行流程的事务	此服务不参与长时间运行流程的事务

如果某个服务参与长时间运行流程的当前事务，那么仅当落实当前事务时，才会保存该服务对事务资源所作的更改。

异步调用

服务始终在自己的事务中运行。要确保异步 SCA 消息的发送参与当前事务，必须将长时间运行流程的 **asynchronousInvocation** 限定符设置为 `commit`。

事务回滚时恢复成功的服务调用

恢复行为取决于所调用服务是否参与当前事务。

某个调用活动调用了参与当前事务的服务。该服务的执行已完成。该服务完成后，如果发生错误，并且该事务回滚到流程在该事务启动前所处的状态，那么还将回滚所调用服务的效果。再次尝试该事务时，将再次调用该服务。

相反，如果所调用服务未参与当前事务，并且该服务返回了响应，那么该响应将存储在独立的事务中。如果存储该响应后发生错误，那么将回滚并重试当前事务。但是，在重试期间不会再次调用该服务，而是恢复所存储的响应并继续进行导航。

业务流程中的故障处理和补偿处理

故障是任何可能导致正常业务流程处理发生变化的异常情况。设计精良的流程应该对故障加以考虑并尽可能对其进行处理。补偿是一种故障处理方式。

相关概念

第 9 页的『活动的状态过渡图』

在活动实例的执行过程中，当遇到重要步骤时，该活动实例的状态将更改。状态和状态过渡随活动类型的不同而有所变化。

故障处理

流程发生故障时，导航将转至故障处理程序。您可以对调用活动、作用域和流程指定故障处理程序。

故障处理程序可以捕获特定名称和/或类型的故障。发生故障时，业务流程管理器将尝试使故障与故障处理程序匹配。它将在外层作用域或发生故障的活动中查找故障处理程序。它使用下列规则来选择故障处理程序：

- 如果该故障没有任何相关联的故障数据，那么业务流程管理器将使用故障名称匹配的故障处理程序。否则，如果有可用的“全部捕获”故障处理程序，那么将使用该故障处理程序。定义了故障变量的故障处理程序无法捕获没有任何数据的故障。
- 如果该故障有相关联的故障数据，那么业务流程管理器将使用故障名称匹配并包含类型与故障数据类型匹配的故障变量的故障处理程序。如果找不到名称和故障数据类型都匹配的故障处理程序，那么将使用不具有故障名称但包含类型与故障数据类型匹配的故障变量的故障处理程序。如果找不到合适的故障处理程序，并且有可用的“全部捕获”故障处理程序，那么将使用该故障处理程序。未定义故障变量的故障处理程序无法捕获带有数据的故障。

如果发生与任何这些故障处理程序定义都不匹配的故障，那么将启动缺省故障处理程序。缺省故障处理程序无需明确指定。缺省故障处理程序将按相应作用域的反向完成

顺序运行紧临内层作用域的所有可用补偿处理程序，并重新抛出该故障给下一层，即，外层作用域或流程。在下一层，业务流程管理器将再次尝试使该故障与可用的故障处理程序匹配。

如果特定故障处理程序以及故障处理程序链中任何位置的“全部捕获”故障处理程序都未捕获该故障，那么该故障将到达流程作用域，并且该流程将结束并处于“已失败”状态。即使某个故障处理程序在流程作用域捕获该故障并对其进行处理，该流程也仍然会结束并处于“已失败”状态。

设计故障处理程序

设计故障处理程序时，请考虑下列选项：

- 捕获故障并尝试解决问题，以使业务流程能够继续进行直至正常完成。
- 捕获故障并确定无法在此作用域解决该故障。在这种情况下，其他可用选项如下所示：
 - 抛出新故障。
 - 重新抛出原始故障，以使另一个作用域可以对其进行处理。
 - 如果这是请求/响应操作，那么通过故障进行应答。
 - 调用人员任务，以解决该问题。如果故障处理程序无法解决该问题，那么可能需要回滚该流程并对其进行补偿。
 - 对于长时间运行流程而言，还可以考虑对该流程使用 **continueOnError** 参数，以便以管理方式处理该故障。

触发故障

可以使用抛出活动和重新抛出活动来触发故障，也可以使用 Java 片段活动通过程序触发故障。

要将故障传播到流程的调用者，请使用指定了故障的应答活动。

使用抛出和重新抛出活动来触发故障

业务流程中的抛出活动可以抛出任何类型的故障，其中包括标准故障，但预期用法模式是抛出业务故障。抛出活动抛出的异常必须在该业务流程中被捕获并处理。如果带有请求/响应接口的流程未处理该流程中的故障，那么该流程将以 **bpws:missingReply** 标准故障结束。对于客户机应用程序而言，此故障在 **StandardFaultException** 对象中返回。

不能通过抛出活动来返回业务故障或标准故障。要将业务故障返回给流程客户机，必须使用应答活动。应答活动只能返回对该流程所实现的接口定义的业务故障。

在故障处理程序中，可以使用重新抛出活动将该故障重新抛出到紧临外层作用域。如果您希望在当前作用域中执行某些故障处理（例如触发特定的补偿处理程序），并且仍希望使外层作用域了解此问题，那么此功能很实用。如果当前故障处理程序无法处理该故障，并且您希望将该故障传播至对某个外层作用域或流程定义的故障处理程序，那么也可以使用重新抛出活动。

由于只能从故障处理程序中重新抛出现有故障，因此只能在故障处理程序中使用重新抛出活动。

通过程序触发故障

通过使用 `raiseFault` 方法，可以在业务流程的 Java 片段中通过程序触发故障。可以通过下列其中一种方法来触发业务故障：

- `raiseFault(QName fault, String variableName);`
- `raiseFault(QName fault);`

以下示例在 `http://process/UpdateCustomerRecordProcess/Interface0/` 名称空间中创建名为 `IncompleteData` 的故障，然后从 Java 片段中抛出此故障。

```
javax.xml.namespace.QName fault = new javax.xml.namespace.QName
    ("http://process/UpdateCustomerRecordProcess/Interface0/", "IncompleteData");
raiseFault(fault);
```

如果所抛出的故障不是对任何 WSDL 接口声明的故障，那么请指定流程的目标名称空间作为该故障的名称空间。然后，可以在业务流程中使用捕获活动来捕获此故障。

请不要直接抛出 `ServiceBusinessException` 对象，而是使用 `raiseFault` 消息来完成此操作。

使用应答活动将故障提供给调用者

指定了故障的应答活动将所指定故障传播至请求/响应操作的调用者。应答活动只能返回对该流程所实现的接口定义的故障。如果业务流程无法正确响应所捕获的故障，但流程发起者可以对其进行响应，那么此功能很实用。例如，如果调用者传递了业务流程找不到的帐号，那么该流程应该通过 `AccountNotFound` 故障应答此服务调用。

指定了故障的应答活动不会使该流程完成并立即返回到调用者。该流程的导航将继续进行，直到进入结束状态为止。

检索关于故障的信息

流程必须能够处理系统故障。要捕获系统故障，可以使用所定义的用于捕获 `runtimeFailure` 标准故障的故障处理程序，也可以使用“全部捕获”故障处理程序。在某些情况下，可能需要使用该故障附带提供的信息。

可以使用下列其中一个构造来检索此信息：

- 在发生标准故障或系统故障时用于存储数据的故障变量。要使用类型化变量来处理故障，您必须手动创建 `StandardFaultType` 复杂类型。
- “全部捕获”故障处理程序。“全部捕获”故障处理程序没有相关联的变量。通过在 Java 片段活动中使用 `getCurrentFaultAsException` 方法，可以从“全部捕获”故障处理程序中检索故障数据。必须将此 Java 片段活动包括在“全部捕获”故障处理程序中。可以使用 `getCurrentFaultAsException` 方法来检索任何类型的故障（而不仅限于系统故障）的数据。

`getCurrentFaultAsException` 方法将该故障作为 `com.ibm.bpe.api.BpelException` 类型的异常对象返回。`BpelException` 对象提供了多个用于获取更多故障信息（例如故障名称）的操作。`BpelException` 对象封装了异常实例。因此，您可以访问故障消息和根异常，如以下示例所示：

```
com.ibm.bpe.api.BpelException bpelexception =
    getCurrentFaultAsException();
System.out.println("Fault Name" +
```

```
bpeexception.getFaultName()  
bpeexception.printStackTrace( System.out);  
Throwable rootCause = bpeexception.getRootCause()
```

“出错时继续”行为

在 WebSphere Integration Developer 中定义流程时，可以对某些活动使用此参数。此参数确定，当那些活动发生故障，并且未定义用于处理该故障的故障处理程序时，如何处理该流程。

如果在活动中检测到故障，那么将开始对该流程进行故障处理。如果紧临外层作用域未处理该故障，并且未设置 **continueOnError** 参数，那么该活动将停止，并且将提供用于继续执行该流程的管理操作。您可以使用业务流程编排器 API 或业务流程编排器资源管理器来查找运行中的包含已停止活动的流程实例。可以重新启动该活动，也可以将其强制完成。如果设置了 **continueOnError** 参数，那么将进行标准故障处理。

例如，可以使用 `forceRetry` 方法对另一输入数据再次启动该活动，也可以使用 `forceComplete` 方法完成该活动并返回输出或故障数据。

```
public interface BusinessFlowManagerService {  
    public void forceRetry(String aaid, ClientObjectWrapper inputMessage,  
        boolean continueOnError);  
    ...  
}
```

如果要在再次出错时停止该流程，那么必须将 **continueOnError** 参数禁用。

使用 `forceComplete` 方法强制完成活动后，将不会再次运行该活动。并且，将使用它的输出消息来继续进行流程导航。

业务流程中的补偿

补偿用于撤销流程中已成功完成的操作。

如果在流程模型中定义了补偿，那么可以通过补偿处理来处理流程实例在运行期间发生的故障。补偿处理将撤销发生故障前已落实的操作的效果，以便恢复到一致的状态。

在流程模型中，可以为长时间运行流程和微流程定义补偿。

长时间运行流程的补偿

长时间运行流程的补偿也称为业务级补偿。可以在作用域级或流程级定义此类补偿。这意味着可以对部分流程或整个流程进行补偿。

补偿由故障处理程序或者作用域或流程的补偿处理程序触发；补偿是流程的另一条导航路径。

外层父作用域接受补偿时，长时间运行流程将自动地对已成功完成的子流程进行补偿。在流程中，将仅对已成功完成的调用活动和作用域活动进行补偿。

微流程的补偿

微流程的补偿也称为技术补偿。当包含微流程的事务或活动会话回滚时，将触发此类补偿。因此，通常对无法通过回滚事务撤销的活动指定撤销操作。当流程实例运行时，将向外层工作单元注册可补偿活动的撤销操作。将根据回滚或落实的结果启动补偿。

如果微流程是可补偿的长时间运行流程的子代，那么该微流程完成后，它的撤销操作可供父流程使用。因此，它可以参与父流程的补偿。对于这些类型的微流程而言，在定义流程模型时，最好对该流程中的所有活动指定撤销操作。

如果补偿处理期间发生故障，那么您必须以手动方式解决补偿操作的问题才能排除故障。可以使用业务流程编排器资源管理器修复这些补偿操作。

相关概念

第 17 页的『微流程的事务行为』

微流程是短时间运行的流程。它们可以在事务中运行，也可以在对该微流程的 SCA 组件指定的活动会话中运行。以下内容对作为事务一部分执行的微流程进行说明。

相关信息

 通过业务流程编排器在流程中使用补偿

从基础结构故障恢复

业务流程管理器提供了用于处理基础结构临时故障的功能。

长时间运行流程跨多个事务。各个事务之间由服务器发送至消息驱动的 Bean 的 Java 消息服务 (JMS) 消息分隔。此 Bean 将入局消息传递给流程服务器以进行处理。每个事务都由下列操作组成：

- 接收请求
- 根据该请求进行导航
- 存储状态
- 发送用于触发后续事务的消息。

服务器可能会由于下列任何一种原因而未能处理消息驱动的 Bean 所接收的消息：

- 无法处理指定数目的连续消息。在这种情况下，假定基础结构不可用。
- 只能处理某些消息。在这种情况下，假定无法处理的任何消息均已损坏。

对这些原因的响应如下所示：

原因	响应
基础结构不可用	在正常处理方式下，消息驱动的 Bean 将根据所指定次数尝试从这种情况恢复。它将尝试保持所有消息在基础结构再次正常运行之前一直可用。此问题的原因可能是数据库故障。
消息已损坏	在尝试所指定次数后，将把该消息放入挂起队列。也可以将该消息从挂起队列移回到输入队列，以再次尝试执行该事务。

业务流程的消息实现如下所示：

- 如果未能处理某条消息，那么将把该消息存储在保留队列中。以后，如果能够成功地处理后续消息，那么将从该队列中自动重放该消息。
- 当消息在保留队列中时，选项如下所示：
 - 如果能够成功处理后续消息，那么将把保留队列中的所有消息移回到消息驱动的 Bean 的输入队列。对于每条消息，将记录该消息被发送到保留队列的次数。如果给定消息的发送次数超出重试限制，那么将该消息放入挂起队列。
 - 如果未能处理下一条消息，那么也将该消息放入保留队列。此过程将一直持续，直到达到保留队列的最大消息数阈值为止。达到此阈值后，消息驱动的 Bean 将把保留队列中的所有消息移至输入队列，然后进入停顿方式。

当消息驱动的 Bean 以停顿方式工作时，它将定期尝试处理消息。将把未能处理的消息放回到挂起队列，而不会增加发送计数或保留队列往返计数。一旦能够成功地处理消息，消息驱动的 Bean 将立即切换回到正常处理方式。

重试限制

重试限制定义将消息放入挂起队列前可以将其传送至保留队列的最大次数。

将一条消息放入保留队列前，未能处理该消息的次数必须达到三次。

例如，如果重试限制为 5，那么在开始最后一次循环前，消息必须进入保留队列 5 次，即失败次数为 $3 * 5 = 15$ 次。如果最后一次重试循环又失败两次，那么将把该消息放入挂起队列。这意味着一条消息被放入挂起队列前，它必须失败 $(3 * \text{重试限制}) + 2$ 次。

在可靠基础结构中运行的性能关键应用程序中，重试限制应该较小，例如 1 或 2。如果重试限制设置为 0，那么重复失败的消息将被重试 3 次，然后它将立即进入挂起队列。

要在管理控制台中指定此业务流程管理器属性，请单击 **服务器** → **应用程序服务器** → **server_name**。然后，在 **容器设置** 部分中，单击 **业务流程编排器容器设置** → **业务流程编排器容器**。在其他属性部分中，单击 **业务流程管理器**。

保留队列消息限制

保留队列消息限制定义保留队列中的最大消息数。如果保留队列溢出，那么系统将进入停顿方式。要使系统在一条消息失败时立即进入停顿方式，请将此值设置为零。要提高业务流程管理器对基础结构故障的承受能力，请增大此值。

要在管理控制台中指定此业务流程管理器属性，请单击 **服务器** → **应用程序服务器** → **server_name**。然后，在 **容器设置** 部分中，单击 **业务流程编排器容器设置** → **业务流程编排器容器**。在其他属性部分中，单击 **业务流程管理器**。

保留队列

保留队列存放已失败的消息，这些消息将被移回到业务流程管理器的内部工作队列以进行重放。消息在失败三次后将进入保留队列。如果消息的失败次数达到 $(3 * \text{重试限制}) + 2$ ，那么它将进入挂起队列。在保留队列达到保留队列消息数限制后，如果又有一条消息失败，那么该队列将溢出，并且系统将进入停顿方式。管理员可以通过查询并回放已失败的消息将此队列中的消息移回到内部队列。

挂起队列

挂起队列包含失败次数达到 (3 * 重试限制) + 2 次的消息。管理员可以通过查询并放回已失败的消息将此队列中的消息移回到内部队列。

重放消息

管理员可以将挂起队列或保留队列中的消息移回到内部队列。可以使用管理控制台或管理命令执行此操作。

停顿方式

保留队列溢出时，将进入停顿方式。发生这种情况时，系统认为发生了严重的基础结构故障，尽管此故障可能是暂时性故障。虽然基础结构故障意味着大部分消息很可能会失败，但是停顿方式的目的是防止系统使用大量资源。在停顿方式下，系统先休眠两秒钟，然后再尝试处理下一条消息。一旦能够成功地处理消息，系统将立即恢复正常的消息处理工作。

人员任务的失败消息处理机制

人员任务管理器既不需要保留队列也不需要重试限制。它只有挂起队列，失败的消息将被放入此队列并从中进行重放。

相关任务

第 259 页的『使用管理控制台查询和重放已失败的消息』

本节描述如何检查和重放未能处理的业务流程或人员任务的消息。

第 276 页的『使用管理命令查询和重放已失败的消息』

使用管理命令来确定是否存在业务流程或人员任务的失败消息并再次尝试处理那些消息。

第 260 页的『刷新失败消息数』

使用管理控制台刷新业务流程或人员任务的失败消息数。

流程的授权与人员分配

授权用于将特定权限授予特定用户或特定用户组。这将确定用户有权对流程和活动执行的操作。

业务流程的授权通过人员任务实现。授权角色用于定义可供特定角色使用的多组操作。相关联的业务流程和活动将继承您对人员任务指定的角色。因此，如果在业务流程中设计了直接插入人员任务，那么该任务的所有者将自动成为活动所有者。每个活动角色都刚好与一个人员任务角色匹配。业务流程管理器使用活动角色进行导航和授权。

业务流程的授权角色

角色是授权级别相同的一组人员。可以对业务流程执行的操作取决于您的授权角色。此角色可以是 J2EE 角色，也可以是基于实例的角色。

相关概念

第 68 页的『人员解析』

人员解析根据一组参数化查询表达式（称为人员分配条件）从人员目录中检索用户信息。

第 56 页的『人员任务的授权角色』

可以对人员任务执行的操作取决于您的授权角色。此角色可以是系统级 J2EE 角色，也可以是基于实例的角色。

业务流程的 J2EE 角色

配置业务流程编排器时，将设置 J2EE 角色。要进行基于 J2EE 角色的授权，必须配置用户注册表并启用全局安全性。

流程支持下列 Java 2 Platform, Enterprise Edition (J2EE) 角色：

- BPESystemAdministrator。分配到此角色的用户拥有所有特权。此角色也称为“业务流程的系统管理员”。
- BPESystemMonitor。分配到此角色的用户能够查看所有业务流程对象的属性。此角色也称为“业务流程的系统监视员”。
- JMSAPIUser。业务流程管理器 JMS API 请求以此角色所映射的用户标识运行，而不考虑调用者。

您可以使用管理控制台来更改对这些角色分配的用户和组。

业务流程和活动的基于实例的角色

系统为流程和活动提供了一组预定义授权角色。设计模型时，可以将这些角色分配给流程和活动。在运行时，通过进行人员解析确定用户与基于实例的角色之间的关联。

流程操作的授权角色

分配给流程角色的人员有权执行下列操作：

角色	有权执行的操作
流程启动者	查看相关流程实例的属性及其输入和输出消息。
流程阅读者	查看相关流程实例的属性及其输入和输出消息。此角色的成员还将自动成为活动以及人员任务活动的相关直接插入待执行任务（包括子任务、后续任务和升级）的阅读者。
流程管理员	管理流程实例、干预已启动的流程以及创建、删除和移交工作项。此角色的成员还将自动成为活动以及人员任务活动的相关直接插入待执行任务（包括子任务、后续任务和升级）的管理员。
流程活动管理员	管理流程的活动。

业务流程管理器使用流程启动者角色来进行流程导航和外部服务调用。如果流程实例仍存在于数据库中，那么请不要从用户注册表中删除流程启动者的用户标识，以使该流程的导航能够继续进行。

要对这些角色分配用户，请使用人员任务。

角色	人员分配
流程启动者	要指定流程启动者，可以将直接插入人员任务分配到流程的始发接收或检取（接收选项）活动。
流程阅读者	要指定流程阅读者，请对流程的相关管理任务设置阅读者角色。该流程中的所有活动都将继承此角色。

角色	人员分配
流程管理员	流程管理员由分配给流程的管理任务定义。该流程中的所有活动都将继承此角色。
流程活动管理员	<p>流程活动管理员由流程的相关管理任务定义。对此任务定义的管理员角色还将被用作流程活动管理员。</p> <p>注： 此管理任务有别于用于确定流程管理员的任务。</p> <p>在流程级定义的活动管理任务是未定义管理任务的活动的缺省管理任务。</p>

活动操作的授权角色

如果设计人员任务并将其作为人员任务活动包括在业务流程中，那么该任务的所有者将自动成为活动所有者。对人员任务定义的角色成员将继承相应人员任务活动的同一角色。业务流程管理器使用活动角色进行导航和授权。直接插入调用任务的潜在启动者是相关接收或检取（接收选项）活动或者事件处理程序的潜在启动者。

活动的基于实例的角色有权执行下列操作：

角色	有权执行的操作
活动阅读者	查看相关活动实例的属性及其输入和输出消息。
活动编辑者	活动阅读者有权执行的操作以及对该活动的消息和其他数据的写访问。
潜在活动启动者	活动阅读者有权执行的操作。此角色的成员可以通过发送消息接收或检取活动。
潜在活动所有者	活动阅读者有权执行的操作。此角色的成员可以声明活动。
活动所有者	处理和完成活动。此角色的成员可以将他们的工作项移交管理员或潜在所有者。
活动管理员	修复由于意外错误而停止的活动以及强制完成长时间运行活动。

流程角色的缺省人员分配

如果未定义某些角色的人员分配条件，或者人员解析失败或未返回结果，那么将以缺省方式进行人员分配。下表列示了应用的缺省值。

业务流程的角色	在流程模型中未定义此角色的情况下 ...
流程管理员	流程启动者成为流程管理员
流程阅读者	没有阅读者

此外，如果未定义用于创建和启动业务流程的调用任务，那么将使用缺省人员分配条件 **Everybody** 作为该流程的潜在启动者。

用于创建和启动业务流程的授权

能够创建和启动流程的用户集由用于创建和启动新流程实例的接收或检取（接收选项）活动的相关调用任务确定，并且也由流程的相关管理任务确定。业务流程将继续承您对这些任务分配的角色。

可以通过下列方式使用人员任务来创建和启动业务流程:

- 对流程的始发接收或检取（接收选项）活动指定直接插入调用任务

某些业务流程可能会更改敏感的业务数据，因此只有经过授权的人员才应该有权创建和启动这些流程。对于此类业务流程而言，可以通过对流程模板指定直接插入调用任务对该流程的始发接收活动指定人员任务。对直接插入调用任务定义的潜在启动者将成为该流程的潜在启动者。

可以通过使用人员任务管理器 API 创建和启动调用任务来启动流程，也可以使用业务流程管理器 API 来启动流程。这两种方法将执行相同的授权检查。如果未指定直接插入任务，那么任何人都可以启动该流程。

- 对流程的始发接收或检取（接收选项）活动指定独立调用任务

还可以使用连接到业务流程的独立调用任务在流程启动时执行授权检查。但是，使用独立调用任务时，请考虑下列事项:

- 仅当流程由调用任务启动时，才会执行授权检查。即，使用业务流程管理器 API 或者直接连接到流程组件的 SCA 客户机来启动流程时，可以不执行检查。
- 它使用 SCA 基础结构来调用流程，而直接插入任务使用内部接口。因此，与独立任务相比，直接插入调用任务的性能更好。
- 无法从人员分配条件定义中访问流程上下文。这意味着独立任务不支持根据流程上下文进行动态人员分配。

- 对流程指定管理任务

流程将继承管理任务的管理员角色。流程管理员可以创建和启动流程。

用于管理业务流程的授权

可以使用管理任务来授权一个用户或一组用户对业务流程及其相关活动执行管理操作。

流程管理

要定义允许哪些用户执行管理操作以及读取流程数据，可以指定一个管理任务作为长时间运行业务流程的组成部分。管理任务的管理员和读者角色确定谁是流程管理员或流程读者。例如，流程管理员可以强制终止流程实例。管理任务与每个业务流程相关联。如果在 WebSphere Integration Developer 中没有为流程设计管理任务，那么在运行时将创建缺省的管理任务。这个缺省任务将流程启动者定义为流程管理员，并且未对流程指定任何读者。

活动管理

可以为每个调用活动或片段活动设计管理任务。此任务确定，除流程管理员以外，还允许谁管理该活动。您还可以在流程级为活动设计缺省管理任务，此任务将应用于每个未指定管理任务的调用活动或片段活动。活动管理任务的管理员角色确定能够管理相应活动的人员。例如，活动管理员和流程管理员可以强制重试该活动。

调用活动可以有相关联的管理任务。对于同步调用活动而言，仅当该活动在调用失败后停止时，才会创建此任务。然后，将使用管理任务来处理修复请求，例如强制完成和强制重试。对于异步调用活动而言，始终创建管理任务。因此，管理员可以在该活动等待异步响应时强制重试或强制完成该活动。

第 2 章 关于人员任务

人员任务是一个组件，它允许人员与服务进行交互。

某些人员任务代表人员的待执行任务。这些任务可以由人员或自动化服务启动。人员任务可以用于实现业务流程中要求人员进行交互的活动，例如以手动方式处理异常以及进行核准。其他人员任务可用于调用服务或者对人员之间的协作进行协调。但是，无论如何启动任务，都由一组人员中被分配该任务的人员执行该任务的相关工作。

您可以通过静态方式将人员任务分配给人员，也可以通过指定将在运行时使用人员目录进行解析的条件（例如角色或组）进行分配。此外，将使用人员任务的输入数据或者业务流程的数据来查找适合于处理任务的人员。

相关任务

第 303 页的第 9 章，『管理业务流程和人员任务』

业务流程和人员任务作为企业应用程序的一部分进行部署和安装。您可以使用管理控制台或管理命令来管理流程模板和任务模板，并可以使用业务流程编排器资源管理器来处理流程实例和任务实例。您可以使用业务流程编排器观察器来创建有关业务流程和人员任务的报告。

第 313 页的『管理任务模板和任务实例』

使用管理控制台或管理命令来管理任务模板。使用业务流程编排器资源管理器来处理任务实例。

第 313 页的『使用管理控制台停止和启动任务模板』

使用管理控制台逐个启动和停止每个已安装的任务模板。

第 314 页的『使用管理命令停止和启动任务模板』

可以使用管理命令来代替使用管理控制台停止和启动任务模板。使用管理命令来停止企业应用程序中的所有任务模板。

第 315 页的『创建和启动任务实例』

您可以根据有权使用的任务模板来创建和启动任务实例。

第 315 页的『处理任务』

要处理某个任务，必须先声明该任务，然后执行完成该任务所需的操作。

第 316 页的『暂挂和恢复任务实例』

可以暂挂任务实例。例如，您可能想暂挂任务实例，以便解决导致该任务实例失败的问题。满足该任务的先决条件后，可以恢复运行该任务实例。

第 317 页的『管理人员任务的优先级』

可以使用人员任务的优先级对任务进行过滤以及对任务列表进行排序。

第 317 页的『管理工作分配』

启动任务后，可能需要管理该任务的工作分配，例如，以更好地在工作组成员之间分配工作量。

第 320 页的『指定离开设置』

如果您打算离开办公室一段时间，那么请指定替代您完成任务的人员。

第 321 页的『指定用户的离开设置』

如果用户无法处理他们的任务，例如休病假，那么可以指定替代该用户执行任务的人员。

第 318 页的『移交您拥有的任务』

如果您是任务所有者，那么可能需要将该任务移交给另一个用户。例如，如果需要他人提供信息才能完成该任务时，就必须这样做。

第 319 页的『移交以您作为任务启动者、发起者或管理员的工作项』

开始执行任务后，可能需要更改工作分配。例如，如果任务所有者正在度假，但必须在此人员归来前完成该任务，那么需要将工作项移交给另一个用户。移交工作项的方式取决于您具有的角色以及任务状态。

第 322 页的『创建工作项』

您可能想为新的潜在所有者创建工作项。例如，在任何当前潜在所有者均无法接受更多的工作时，就可能需要执行此操作。此外，如果对人员目录执行的查询未返回任何潜在所有者，那么也可能需要创建工作项。例如，在长时间运行流程中，如果公司在流程启动后发生更改，那么可能会发生找不到潜在所有者的情况。

第 323 页的『删除工作项』

您可能想删除工作项。例如，如果错误地创建了工作项，或者工作项的所属用户已离职，那么可以执行此操作。

第 323 页的『查看任务升级』

升级将用户可能无法及时完成所分配任务这一情况通知升级接收者。

第 324 页的『发送升级电子邮件』

当任务逾期时，它可能会引起升级。可以将系统设置为向指定的人员发送电子邮件，以便将关于该升级的信息通知那些人员。

任务模板

人员任务模板包含使用 WebSphere Integration Developer 创建的已部署任务模型的定义，或者包含在运行时使用业务流程编排器 API 创建的任务的定义。

模板包含属性（例如任务名称和优先级）并整合工件（例如升级模板、定制属性和人员查询模板）。除了设计任务模板时指定的属性以外，已安装的任务模板还可以处于下列其中一种状态：

已启动 启动任务模板后，可以启动该模板的新实例。

已停止 要卸载人员任务应用程序，必须先停止任务模板。当任务模板处于“已停止”状态时，无法启动此模板的新实例。

在运行时，可以通过创建 `com.ibm.task.api.TaskModel` 类的实例来进行待执行任务或协作任务建模。然后，可以使用这些实例来创建可复用的任务模板或者直接创建一次性的任务实例。在运行时进行的人员任务设计工作基于 Eclipse 建模框架（EMF）。

相关任务

第 415 页的『在运行时创建任务模板和任务实例』

通常，使用建模工具（例如 WebSphere Integration Developer）来构建任务模板。然后，将任务模板安装到 WebSphere Process Server 中并根据这些模板创建实例，例如使用业务流程编排器资源管理器来创建实例。但是，也可以在运行时创建人员任务实例/模板或参与任务实例/模板。

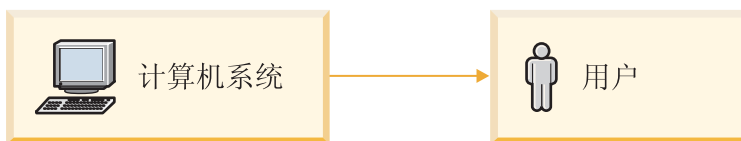
人员任务的类型

任务类型从设计期间指定的任务模板类型派生。

人员任务的类型如下所示：

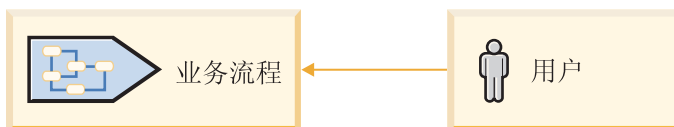
待执行（参与）任务

支持 Web Service 与人员之间的交互，此交互使人员能够实现服务。例如，待执行任务可以是业务流程中的人员任务活动。



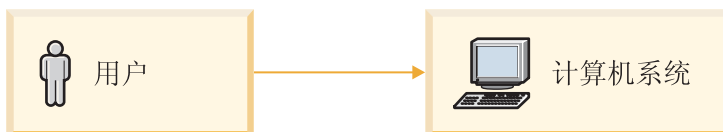
管理任务

管理任务由管理员使用，用于解决流程中发生的技术问题。目前，管理任务只能用于业务流程。



调用（发端）任务

支持人员与 Web Service 之间的交互，这些交互使人员能够创建和启动服务。例如，用户可以启动业务流程或者通过调用任务向该业务流程发送事件。



协作（纯人员）任务

支持人员与人员之间的交互，这些交互使人员能够以结构化的受控方式与其他人员分担工作。协作任务不直接与业务流程或其他 Web Service 进行交互。



对人员任务进行版本控制

如果您希望同一个人员任务的多个备用实例在运行时环境中共存，请使用版本控制功能。

在 WebSphere Integration Developer 中设计独立人员任务时，可以指定版本控制信息。任务模板的版本由其生效日期确定。这意味着，不同版本的模板可以具有相同的任务模板名称，但它们的生效日期不同。在运行时使用的任务模板版本由该任务是用于早期绑定方案还是后期绑定方案决定。

早期绑定

在早期绑定方案中，将在建模期间或者部署任务模板时确定所使用的任务版本。主调组件根据服务组件体系结构（SCA）连接来调用以静态方式绑定的专用任务模板。即使存在另一个根据其生效日期确定有效的任务模板版本，也将使用当前以静态方式连接的任务模板并忽略所有其他版本。

SCA 连接是早期绑定的一个示例。如果将独立引用连接到某个人员任务组件，那么使用此引用进行的每次任务模板调用都将面向该人员任务组件所代表的特定版本进行。

后期绑定

在后期绑定方案中，将在创建任务实例时确定所使用的人员任务模板。在这种情况下，将使用当前有效的任务模板版本。新版本的模板将取代所有先前版本的模板。现有任务实例将继续使用启动时与其相关联的任务模板运行。因此，任务模板分为下列三类：

- 用于新任务实例的当前有效任务模板
- 仍适用于运行中的任务实例但已不再有效的任务模板
- 根据生效日期，未来将会生效的任务模板

在业务流程编排器资源管理器中调用新任务是后期绑定的一个示例。创建的实例始终基于具有过去生效日期的最新任务模板版本。后续任务和子任务始终使用后期绑定进行调用。

任务实例

任务实例是任务模板在运行时的表现。

独立任务与直接插入任务

面向服务的体系结构（SOA）模式提供了一组松散耦合组件的软件解决方案实现建议。遵循 SOA 模式的人员任务称为独立任务，而作为业务流程组成部分定义的人员任务称为直接插入任务。

下表列示了可用于独立任务和直接插入任务的任务类型：

表 3.

实现	调用任务	待执行任务	协作任务	管理任务
独立	是	是	是	否
直接插入	是	是	否	是

独立任务

独立任务遵循 SOA 模式，它们与调用它们的组件（待执行任务）或者它们调用的组件（调用任务）松散耦合。它们可以通过服务组件体系结构（SCA）基础结构连接到另一个组件。

它们仅通过 SCA 方法与伙伴组件进行通信。即，待执行任务接收输入消息并返回输出消息或故障消息，调用任务发送输入消息并接收输出消息或故障消息。除此之外，不执行任何其他信息交换或生命周期控制操作。

由于独立任务是单独设计的，因此可以重复使用。独立任务始终将它们的公共事件基础结构（CEI）事件和审计日志事件作为人员任务管理器事件发出。

独立任务以下列方式作为 SCA 组件：

- 待执行任务带有可以连接到客户机组件的接口。
- 调用任务带有可以连接到所要调用的服务的引用。
- 协作任务是独立的 SCA 组件。虽然协作任务是独立任务，但它们有两个人员接口，因此无法连接到服务组件。

管理任务既不是独立任务也不是 SCA 组件。

对于与业务流程配合使用的独立任务而言，下列规则适用：

- 它们的生命周期与流程无关。
 - 缺省情况下，待执行任务由流程创建，它们随着该流程的删除而被删除。
 - 调用任务可以创建流程。但是，它们不会随着该流程的删除而被删除，因此任务结果可视。
- 待执行任务可以被视为业务流程中的调用活动。
- 调用任务连接到接收或检取（接收选项）活动或者 on-event 事件处理程序。
- 任务描述、显示名和文档同时支持多种语言。
- 独立任务无法访问流程上下文。它们无法访问其他流程活动中的流程变量、定制属性或数据。

如果协作任务是顶层任务，那么它的生命周期是独立管理的，它将被手动删除或者在达到所指定时间后被自动删除。如果协作任务是子任务或后续任务，那么它的生命周期由它的父任务或顶层任务管理。

直接插入任务

直接插入任务是业务流程的组成部分。直接插入任务可以是待执行任务、调用任务和管理任务。由于协作任务利用人员之间的交互并且不直接与流程进行交互，因此它们不能是直接插入任务。直接插入任务既不能被看作 SCA 组件（不能进行连接），也不能在其他流程或活动中重复使用。

直接插入任务能够访问流程上下文和流程数据，例如流程变量、定制属性和活动数据。对于涉及划分职责的任务而言，此功能很实用。直接插入待执行任务将它们的 CEI 事件和审计日志事件作为业务流程管理器人员任务活动事件发出。它们的子任务和后续任务将事件作为人员任务管理器事件发出。

对于直接插入任务，下列规则适用：

- 待执行任务是流程中的人员任务活动。它们共享同一种状态，但人员任务活动未反映任务子状态。
- 调用任务与接收或检取（接收选项）活动或者 on-event 事件处理程序相关联。
- 管理任务连接到流程或者流程中的活动。
- 生命周期通常由流程确定。
 - 待执行任务和管理任务由业务流程创建，并且与流程一起被删除。
 - 如果调用任务由业务流程创建和启动，那么它们的生命周期由该流程确定，并且与该流程一起被删除。如果它们是通过人员任务管理器 API 创建并启动的，那么它们的生命周期与流程无关，它们的结果即使在该流程被删除后也仍可以显示。
- 待执行任务和调用任务的描述、显示名以及文档只支持一种语言。
- 直接插入任务没有“有效时间”。但是，可以对待执行任务的相应人员任务活动定义到期时间。
- 只有直接插入调用任务具有“删除持续时间”，但仅当此任务由人员任务管理器 API 启动时，这属性才适用。
- 对直接插入任务执行的更新操作只支持一小部分任务属性。只能更新在流程或活动中没有相应表示的任务属性。有关 update 方法的更多信息，请参阅 com.ibm.task.api 包中 HumanTaskManager 接口的 Javadoc。

直接插入任务用于进行流程授权:

- 待执行任务的角色读者、管理员、潜在所有者、所有者和编辑者与流程中人员任务活动的相应角色完全相同。
- 直接插入调用任务的潜在启动者角色确定，哪些人员能够调用相应的接收或检取（接收选项）活动或者 on-event 事件处理程序以及将消息发送至那些活动或处理程序。请注意，潜在启动者和潜在实例创建者角色具有完全相同的人员分配。如果未定义某个直接插入调用任务，那么每个人都有权启动该活动或事件处理程序。
- 流程管理任务的管理员和读者角色确定流程管理员或流程读者。例如，流程管理员可以强制终止流程实例。
- 活动管理任务的管理员角色确定能够管理相应活动的人员。例如，活动管理员和流程管理员可以强制重试该活动。
- 流程读者和流程管理员权限将由每个流程活动或直接插入人员任务继承。

人员任务与业务流程的关系

调用任务可以与接收或检取（接收选项）活动或者 on-event 事件处理程序相关联。这些任务可以是直接插入任务或独立任务。如果您正在使用业务流程管理器 API，那么只有直接插入调用任务可以影响用于调用接收活动的授权。缺省情况下，每个人都能够发送消息给接收活动、检取活动或 on-event 事件处理程序。这包括在启动接收活动时调用业务流程。

管理任务与每个业务流程相关联。管理任务确定有权管理和阅读流程的人员。如果未在 WebSphere Integration Developer 中设计流程的管理任务，那么在运行时将创建缺省的临时任务。此缺省任务将流程启动者定义为流程管理员，并且未对流程指定读者。

可以为每个调用活动或片段活动设计管理任务。此任务确定，除流程管理员以外，还允许谁管理该活动。并且，还可以设计缺省活动管理任务，此任务将应用于每个未明确指定管理任务的调用活动或片段活动。

调用活动可以有相关联的管理任务。对于片段活动和同步调用活动而言，仅当该活动在调用失败后停止时，才会创建此任务。然后，将使用管理任务来处理修复请求，例如强制完成和强制重试。对于异步调用活动而言，始终创建管理任务。因此，管理员可以在该活动等待异步响应时强制重试或强制完成该活动。

独立待执行任务可以实现异步调用活动。这些活动也有相关联的管理任务。直接插入待执行任务实现人员任务活动。在运行时，将为这些活动创建管理任务。

相关概念

第 45 页的『人员任务的生命周期』

人员任务支持人员与 Web Service 或业务流程进行交互。在任务生存期内可以执行的交互取决于该任务是待执行任务、协作任务、调用任务还是管理任务。某些交互只能在任务处于特定状态时进行，而这些交互又会影响人员任务的状态。

第 29 页的『业务流程和活动的基于实例的角色』

系统为流程和活动提供了一组预定义授权角色。设计模型时，可以将这些角色分配给流程和活动。在运行时，通过进行人员解析确定用户与基于实例的角色之间的关联。

子任务

当人员需要将分配给他们的部分工作委托他人完成，但希望继续对整体结果进行控制时，可以借助于子任务。子任务还可用于调用支持服务，以帮助人员完成所处理的任

务。可以根据业务流程编排器数据库中存储的独立任务模板、根据在运行时创建的任务模板或者通过在运行时提供新任务模型来创建子任务。父任务可以是待执行任务或协作任务，并且它的 **supportsSubtask** 属性必须设置为 **true**。所创建的子任务可以是协作任务或调用任务。这些子任务可以有子任务或后续任务。

对输入消息类型或输出消息类型没有限制。但是，子任务的启动者必须提供输入消息。子任务完成后，父任务的所有者可以将子任务输出数据映射至父任务的输出消息。

授权注意事项

除了在子任务启动时对其指定的内容以外，子任务还将从它的父任务继承授权角色：

- 父任务的读者、编辑者、发起者和所有者成为子任务及其升级的读者
- 父任务的管理员成为子任务的管理员
- 父任务的升级接收者成为子任务的读者

生命周期注意事项

第一个子任务启动时，父任务将进入“正在等待子任务”子状态。它将一直处于这种子状态，直到最后一个子任务进入其中一种结束状态（已完成、已失败、已到期或已终止）为止。父任务的某些生命周期操作（状态更改）将传播到它的子任务。因此，当父任务暂挂、恢复、终止、被删除或到期时，它的所有子任务也将暂挂、恢复、终

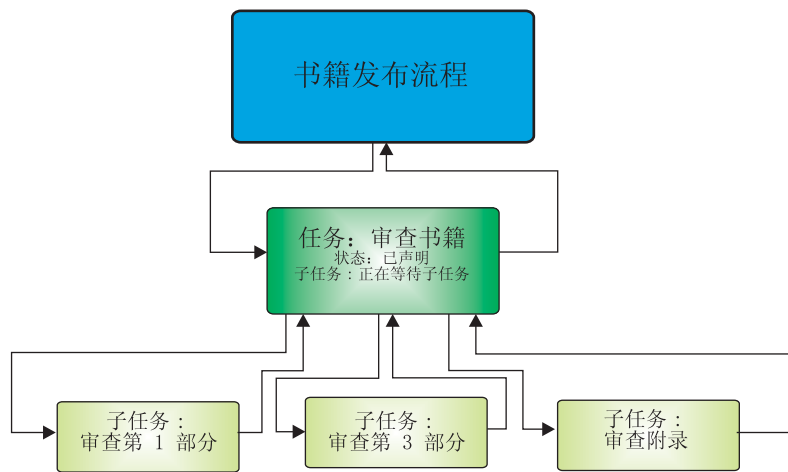
止、被删除或到期。父任务的“已升级”子状态不传播；父任务升级时，子任务不升级。子任务有它们自己的升级，并且，仅当它们自己的某个升级被触发时，才会设置它们的“已升级”子状态。

对子任务执行的某些生命周期操作将与父任务的生命周期操作发生冲突，因此不允许执行这些操作。这些操作主要是将会对子任务生命周期的结束产生影响的操作，它们必须与父任务协调。可以对子任务执行下列操作：

- 始终支持不会与父任务发生冲突的生命周期操作。这些操作包括声明、取消声明、完成、创建以及启动子任务或进一步的后续任务。
- 子任务可以到期。
- 子任务可以因为其工作要求停止而被暂挂和恢复（尽管父任务的工作将继续进行）。
- 子任务可以被终止。
- 子任务可以有自己的升级，以使父任务所有者和子任务发起者可以更好地控制子任务的进度。
- 作为子任务启动的任务的自动删除设置将被忽略。子任务将随着它们的父任务的删除而被删除。不支持使用业务流程编排器 API 来删除各个子任务。

示例：父任务与协作任务之间的交互

下图显示了一个书籍发布流程，在此流程中，人员任务活动有子任务。



在书籍发布流程中，“审查书籍”任务由 Linda 声明。她意识到独自审查此书对于她来说工作量太大，并意识到需要专门的知识才能审查此书的某些章节。她决定不采用标准发布流程，而是将部分任务委托一些同事完成。因此，她根据“审查书籍章节”模板创建三个附加任务：“审查第 1 部分”、“审查第 3 部分”和“审查附录”。Linda 自己审查此书的第 2 部分。

她将整本书作为子任务的输入以使她的同事有足够的上下文信息，但对任务描述添加了注释，以通知各位同事只需审查该书的指定部分。她将这些任务分配给她的同事：John 审查第 1 部分，Cindy 审查第 3 部分，Mary 审查附录。然后，她启动三个任务作为她自己的“审查书籍”任务的子任务。她的先前处于“已声明”状态的任务将进入“正在等待子任务”子状态，直到全部三个子任务都完成为止。

Cindy、John 和 Mary 声明他们的子任务并开始审查所负责的书籍部分。与此同时，Linda 审查该书的第 2 部分。当她完成所负责的审查部分后，她检查各位同事的进度。Cindy 和 John 已完成了审查工作，但 Mary 仍在审查篇幅较大的附录部分。Linda 的任务仍处于“正在等待子任务”状态。虽然 Linda 无法完成她的任务，但她可以开始根据 Cindy 和 John 的子任务输出合并审查意见。

与此同时，Mary 也完成了她的子任务，因此 Linda 的“审查书籍”任务将离开“正在等待子任务”子状态。现在，Linda 将 Mary 的审查意见与该书其余部分的审查意见合并，从而完成她的任务。书籍发布流程将继续进行。由于“审查书籍”任务是直接插入人员任务，因此，它将在业务流程实例被删除时与它的子任务一起被删除。

示例：父任务与调用任务之间的交互

父任务与调用任务之间的交互类似于父任务与协作任务之间的交互。任务所有者将根据现有的调用任务模板创建任务，然后将其作为她自己的任务的子任务启动。父任务将进入“正在等待子任务”子状态并等待调用子任务返回。所有子任务完成后，父任务将离开“正在等待子任务”子状态并可以完成。

相关概念

第 56 页的『人员任务的授权角色』

可以对人员任务执行的操作取决于您的授权角色。此角色可以是系统级 J2EE 角色，也可以是基于实例的角色。

第 45 页的『人员任务的生命周期』

人员任务支持人员与 Web Service 或业务流程进行交互。在任务生存期内可以执行的交互取决于该任务是待执行任务、协作任务、调用任务还是管理任务。某些交互只能在任务处于特定状态时进行，而这些交互又会影响人员任务的状态。

后续任务

当用户希望将分配给他们的部分工作委托他人完成并移交对完成工作的控制权时，可以借助于后续任务。

可以根据业务流程编排器数据库中存储的独立任务模板、根据在运行时创建的任务模板或者通过在运行时提供新任务模型来创建后续任务。后续任务可以有自己的后续任务，从而形成任务链。

后续任务只能是协作任务。您可以从待执行任务或者 **supportsFollowOnTask** 属性设置为 **true** 的协作任务中启动协作任务。

后续任务的输入消息类型可以与前趋任务不同。如果后续任务的输入消息类型与前趋任务相同，那么前趋任务的输入消息内容将自动传递到后续任务。在创建或启动后续任务时，可以覆盖消息内容。

对于后续任务链而言，每个后续任务的输出消息类型和故障消息类型必须与该链中的顶层任务完全相同，这是因为，链中最后一个后续任务会将该消息返回给主调组件或人员（发起者）。父任务的输出消息内容或故障消息内容将始终被复制到后续任务的输出消息或故障消息。在后续任务中，可以对这些消息进行修改。

授权注意事项

后续任务继承前趋任务的授权角色:

- 前趋任务的阅读者、编辑者、发起者和所有者成为后续任务及其升级的阅读者
- 前趋任务的管理员成为后续任务的管理员
- 前趋任务的升级接收者成为后续任务的阅读者

生命周期注意事项

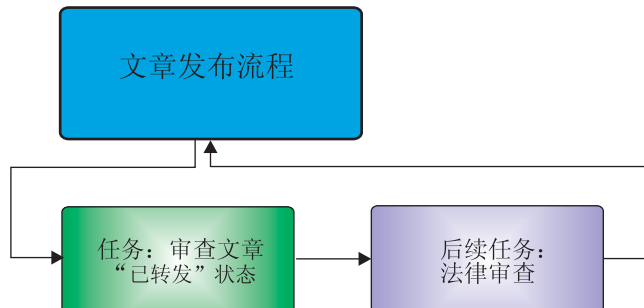
后续任务启动时，前趋任务将进入“已转发”状态。后续任务是其前趋任务的子代，因此前趋任务的某些生命周期操作（状态更改）将传播到后续任务。当前趋任务暂挂、恢复、升级、终止、被删除或到期时，它的所有后续任务也将暂挂、恢复、升级、终止、被删除或到期。

对后续任务执行的某些生命周期操作将与前趋任务的生命周期操作发生冲突，因此不允许执行这些操作。这些操作主要是将会对后续任务生命周期的结束产生影响的操作，它们必须与前趋任务协调。可以对后续任务执行下列操作:

- 始终支持不会与父任务发生冲突的生命周期操作。这些操作包括声明、取消声明、完成、创建以及启动子任务或后续任务。
- 由于后续任务链对于主调组件或人员（发起者）而言就象是单个任务，因此后续任务不支持“有效时间”，而是将在链中顶层任务的到期计时器结束时到期。
- 后续任务可以因为其工作要求停止而被暂挂和恢复（尽管父任务的工作将继续进行）。
- 后续任务可以被终止。
- 后续任务可以有自己的升级，以使前趋任务所有者和后续任务发起者可以更好地控制该后续任务的进度。
- 不支持使用业务流程编排器 API 来删除各个后续任务。

示例: 后续任务

下图显示了一个发布流程，在此流程中，人员任务活动有后续任务:



在文章发布流程中，“审查文章”任务由 John 声明。此流程还授权 John 在法律方面对文章进行审查与核准。但是，此文章描述与竞争者的产品进行协作，因此从法律的角度看非常敏感。John 检查了此文章中信息方面的内容，并决定将此文章转交法律部门的

Sarah 以进行进一步审查。他创建了“法律审查”任务，并提供了描述内容以概括对法律方面的关注。他将此文章包括在任务的输入中，然后将其分配给 Sarah。然后，他启动一个新任务作为他自己的“审查文章”任务的后续任务。他的任务进入“已转发”状态，该任务的工作结束。此流程将等待所调用的“审查文章”任务的响应。

Sarah 声明“法律审查”后续任务并开始在法律方面进行审查。她作了一些评论并完成了她的任务。后续任务的输出消息将传递到业务流程。文章发布流程使用“审查文章”任务的相关输出继续进行，但该输出实际上来自于“法律审查”后续任务。由于“审查文章”任务是直接插入人员任务，因此，它将在业务流程实例被删除时与“法律审查”任务一起被删除。

升级

升级是指人员任务在所指定时间内未被处理时自动生成的警报。例如，如果在定义的时间限制内未声明或未完成任务，那么将生成此警报。对于一个任务，可以指定一个或多个升级。这些升级可以并行启动，也可以作为一连串升级启动。

通过对任务模板定义升级模板，或者通过在运行时对临时任务定义升级模板，可以为任何任务定义升级。

升级在任务处于特定状态时被激活，并且，仅当任务在升级时间限制到期前未进入预期任务状态（监视状态）时，才会进行升级。升级超时时间限制以字符串格式指定，并且由您对该任务指定的日历解释。您可以指定多个具有相同激活状态的升级，即升级链。

可以定义当任务进入下列任务状态时激活的升级：

就绪 对于处于“就绪”状态的任务，可以为下列情况定义升级：

- 如果未及时使用预期任务状态“已声明”来声明任务，那么进行升级。
- 如果未及时使用预期任务状态“已结束”来完成任务，那么进行升级。

已声明 对于处于“已声明”状态的待执行任务或协作任务，可以为下列情况定义升级：

- 如果未及时使用预期任务状态“已结束”来完成任务，那么进行升级。
- 如果未及时使用预期任务状态“子任务已完成”来完成此任务的子任务，那么进行升级。此外，还可以使用“正在等待子任务”激活状态作为预期状态，以便跟踪子任务的进度。

正在等待子任务

当待执行任务或协作任务处于“正在等待子任务”子状态时，如果未及时使用预期任务状态“子任务已完成”来完成此任务的子任务，那么将此任务升级。

正在运行

当调用任务处于“正在运行”状态时，如果被调用服务未及时使用预期任务状态“已结束”返回，那么进行升级。

可以定义重复的升级。这些升级在每次发生超时情况时检查同一种预期任务状态并执行所定义的升级操作，直到任务进入预期状态为止。

发生升级时，受升级影响的人员（升级接收者）将接收到工作项。根据升级定义的不同，升级接收者还可能会接收到通知他们任务已升级的电子邮件。所要通知的一组用户由人员查询定义。此查询必须解析为一组用户标识，以便创建工作项。

通过使用 `increasePriority` 属性，可以定义升级以提高所升级任务的优先级。可以在第一次进行迭代时自动提高优先级，也可以在每次进行升级迭代时提高优先级。

升级生命周期

升级在其生命周期内将进入下列状态：

- 在创建后，升级处于不活动状态，直到任务进入激活状态为止。
- 当任务进入升级的激活状态后，该升级将进入“正在等待”状态。计时器将启动，升级将等待计时器超时。
- 发生超时后，将检查所监视任务的 `atLeastExpectedState` 属性。如果该任务已进入或离开此状态，那么该升级将进入“不适用”状态。如果尚未进入预期状态，那么该升级将进入“已升级”状态，并且将调用所指定的升级操作。

创建升级后，不能对其进行修改。可以重复地执行升级操作。重复时间间隔由该升级的 `autoRepeatDuration` 属性定义。

链接的升级

当任务进入升级链中第一个升级的启动状态时，将激活一连串升级。升级链中所有升级的激活状态必须相同。在升级链中，除始终处于活动状态的重复升级以外，每次只能有一个升级处于活动状态。对定义成序列的多个升级的处理按顺序进行：第一个升级发生时，将激活升级链中的下一个升级，依此类推。

所链接升级的等待持续时间相对于上一个升级的超时计算，而不是相对于任务进入升级激活状态的时间计算。因此，如果升级链中第一个升级的等待持续时间为 2 小时，并且该链中第二个升级的等待持续时间为 3 小时，那么第一次超时在任务进入激活状态 2 小时后发生，第二次超时在第一次超时发生 3 小时后发生，也就是在任务进入激活状态 5 小时后发生。这种行为确保升级链中的后续升级不会在前趋升级超时前超时。

升级的动态持续时间

对于某些升级而言，可以在运行时动态地设置升级时间段。为此，请在定义该升级时指定替换表达式来代替固定值。必须将持续时间变量括在百分号 (%) 中。

此变量可以是下列任何一项：

- 任务变量，例如 `%htm:input.myEscalationDurationValue%`
- 定制属性，例如 `%htm:task.property.myEscalationDurationValue%`
- 流程变量，例如 `%wf:variable.myVariable\myPart\myEscalationDurationValue%`

您必须确保所访问的上下文数据在进行升级评估时可用。

下表说明对升级持续时间进行评估的时机：

以下项目的持续时间	评估时机	在任务进入以下状态前，必须设置上下文日期：
升级	任务进入升级的激活状态时	升级的任务激活状态
重复升级	发生升级时	已升级

相关概念

第 68 页的『人员目录』

人员目录存储用于进行人员解析的用户信息。

相关任务

第 495 页的『创建通知事件处理程序』

当人员任务升级时，将生成通知事件。业务流程编排器提供了用于处理升级的功能，例如创建升级工作项或发送电子邮件。您可以创建通知事件处理程序，以便对处理升级的方式进行定制。

升级的电子邮件通知

发生升级时，受升级影响的人员将接收到工作项。此外，您还可以向他们发送一封电子邮件，以便通知他们任务已升级。电子邮件有一些相应的规则。

每个升级都可以有各自的电子邮件。您可以对标准升级电子邮件进行个性化，例如，使其符合贵公司的标准。要对电子邮件进行个性化，请在人员任务编辑器中编辑升级详细信息。

电子邮件的主题行和正文都可以包含替换表达式，以强化该电子邮件与所引用任务之间的关系，例如包含任务名的替换表达式。必须在发送电子邮件前设置这些表达式，否则电子邮件收件人在他们的电子邮件中将看到 *%variable name%*。您可以使用任何任务和升级表达式。

以下 HTML 片段提供了包含替换表达式的样本电子邮件：

```
<html>
<head>
</head>
<body lang="EN-US"><div>
<p>The task '<span style="font-size:14.0pt">%htm:task.displayName%/span></b>'
with id '<span style="font-size:14.0pt">%htm:task.instanceID%/span></b>
'&nbsp;has been escalated because the </p>
<p>expected state '<span style="font-size:14.0pt">%htm:escalation.expectedTaskState%/span>
</b>'
&nbsp;has not been reached in time.
</p>
<p>&nbsp;</p>
<p>The task has the following description: </p>
<p><span style="font-size:14.0pt;color:red">%htm:task.description%/span></p>
<p>&nbsp;</p>
<p><span style="font-size:14.0pt;color:green">The name of the Escalation is: %htm:escalation.displayName%
and the escalation description reads: %htm:escalation.description%/span></p>
<p>&nbsp;</p>
<p><a href="%htm:task.URLPrefix?id=%htm:task.instanceID%">Task details</a></p>
<p><a href="%htm:escalation.URLPrefix?id=%htm:escalation.instanceID%">Escalation details</a></p>
</div>
</body>
</html>
```

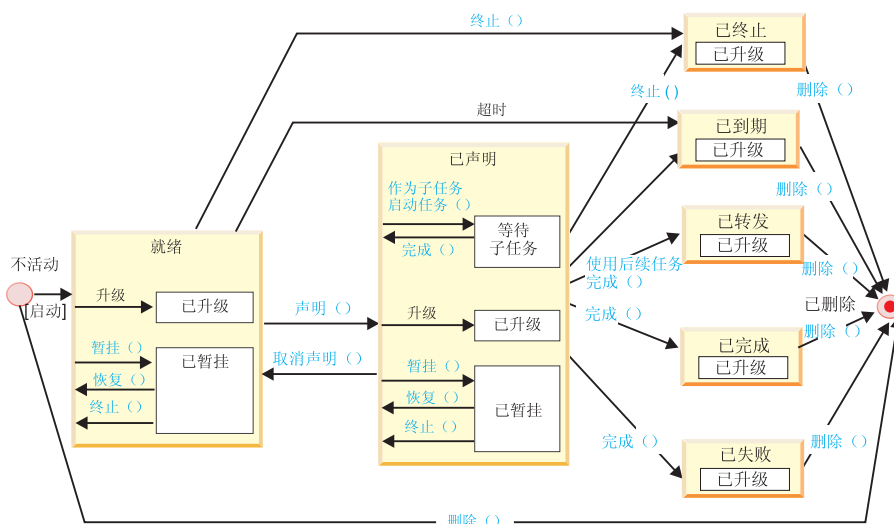
人员任务的生命周期

人员任务支持人员与 Web Service 或业务流程进行交互。在任务生存期内可以执行的交互取决于该任务是待执行任务、协作任务、调用任务还是管理任务。某些交互只能在任务处于特定状态时进行，而这些交互又会影响人员任务的状态。

待执行任务和协作任务

待执行任务和协作任务支持人员在业务流程中执行工作（直接插入任务）或实现公用 Web Service（独立任务）。待执行任务和协作任务在启动方式上有所不同。待执行任务由客户机应用程序或主调组件自动创建。协作任务由个人创建和启动。

下图说明待执行任务和协作任务的生命周期内可能出现的状态过渡：



创建任务后，它将进入不活动状态。在这种情况下，您可以更新任务属性或设置定制属性，但无法声明该任务。要处理待执行任务或协作任务，必须将其启动。

任务启动后，它将进入“就绪”状态。在这种情况下，此任务将等待其中一个潜在所有者声明此任务并执行它的相关联工作。在此状态下，可能会发生下列异常事件：

- 此任务可能会因为未被及时声明而升级。它将进入“已升级”子状态，并且将在任务生命周期的余下时间内一直处于这种子状态。
- 此任务可能会被手动暂挂。它将进入“已暂挂”子状态。在此状态下，对此任务执行的大部分操作将被阻塞。此任务可以被手动恢复，也可以由您对暂挂操作设置的计时器自动恢复。
- 此任务可能会到期。此状态更改将使此任务结束。
- 您可以使用终止操作手动终止此任务。此状态更改将使此任务结束。

在正常的任务流中，其中一个潜在所有者声明此任务并成为所有者。此任务将进入“已声明”状态，所有者和编辑者可以对其进行处理。当任务处于“已声明”状态时，任务所有者可以执行下列操作：

- 如果他们需要获得支持才能完成工作，那么可以使用子任务将部分工作委托他人完成。这些子任务可以是协作任务或调用任务。接着，父任务将进入“正在等待子任务”子状态并一直处于此状态，直到所有子任务进入结束状态为止。父任务在等待子任务时，他们可以将其暂挂，但是，不能完成此任务，也不能取消声明。
- 如果他们希望委托他人完成工作，那么可以创建协作任务并通过后续任务完成任务。父任务将进入“已转发”结束状态。

- 如果他们希望让他人接管任务的整体职责，那么可以将所有者工作项移交给另一个潜在所有者或管理员。
- 如果他们希望放弃任务的所有权，那么可以取消声明此任务。此任务将再次进入“就绪”状态，并且可以由其中一个潜在所有者声明。

在“已声明”状态下，可能会发生下列异常事件：

- 此任务可能会因为未及时完成或者等待子任务的时间过长而升级。它将进入“已升级”子状态，并且将在任务生命周期的余下时间内一直处于此状态。
- 此任务可能会被手动暂挂。它将进入“已暂挂”子状态。在此状态下，对此任务执行的大部分操作将被阻塞。此任务可以被手动恢复，也可以由您对暂挂操作设置的计时器自动恢复。并且，当计时器到期时，对此任务的声明将被取消，此任务将再次进入“就绪”状态。
- 此任务可能会到期。此状态更改将使此任务结束。
- 您可以使用终止操作手动终止此任务。此状态更改将使此任务结束。

当所有者处理任务完毕后，他们将完成此任务。于是，如果此任务成功完成，那么它将进入“已完成”状态；如果发生错误，那么它将进入“已失败”状态。

在“已失败”、“已终止”、“已完成”和“已到期”结束状态下，无法执行工作。如果任务模板指定了自动删除，那么此任务将被立即删除或者在删除计时器到期后被删除。如果不进行自动删除，那么此任务将一直处于结束状态，直到被明确删除为止。父任务被删除时，它的子任务也将被删除。如果任务处于“已转发”结束状态，那么自动删除功能不适用。在这种情况下，父任务将与后续任务一起被删除。删除计时器在后续任务进入结束状态时启动。

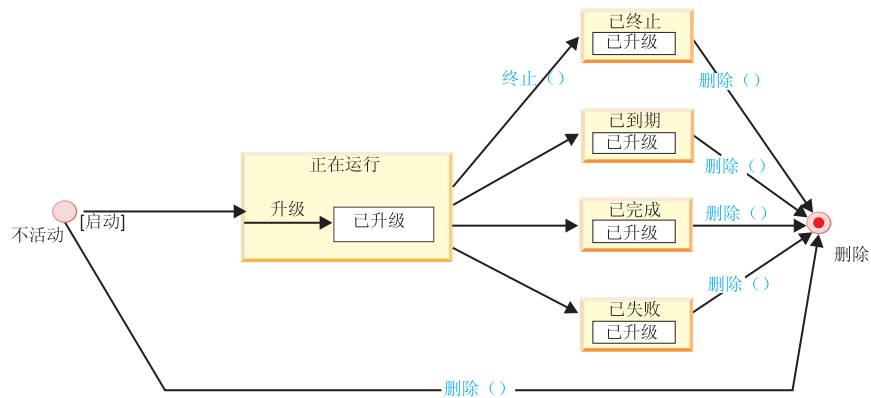
还有其他一些规则适用于直接插入待执行任务。直接插入任务是业务流程的组成部分，因此它们的生命周期由流程生命周期控制：

- 此任务由业务流程隐式地创建和启动。
- 在业务流程中，此任务由人员任务活动表示。此任务和活动处于相同的状态，例如，当此任务处于“就绪”状态时，人员任务活动也处于“就绪”状态。人员任务活动未反映“已转发”状态或任务子状态。
- 即使直接插入任务有子任务，人员任务活动也不会了解到这种情况，它将处于“已声明”状态并等待父任务完成。
- 即使直接插入任务有后续任务，人员任务活动也不会了解到这种情况，它将处于“已声明”状态并等待后续任务完成。
- 直接插入待执行任务不具有“有效时间”，并且不能被手动终止。到期和终止都由人员任务活动或业务流程控制。
- 这些任务将与业务流程一起被删除。它们既不能被手动删除，也不具有“删除持续时间”。

调用（发端）任务

调用任务支持人员调用 Web Service。创建和启动调用任务的人员将成为任务发起者。此任务启动后，它将自动调用该服务并等待结果。当服务结果可用时，调用任务将存储该结果；只要此任务存在，发起者就可以检索该结果。

下图说明调用任务的生命周期内可能出现的状态过渡：



任务在创建后将进入不活动状态。在此状态下，您可以更新任务属性或设置定制属性。要调用该服务，必须启动此任务。此任务可以由发起者或其中一个潜在启动者启动。

任务启动后，它将进入“正在运行”状态。在此状态下，此任务将等待所调用的服务返回。在此状态下，可能会发生下列异常事件：

- 此任务可能会因为该服务未及时返回而升级。它将进入“已升级”子状态，并且将在任务生命周期的余下时间内一直处于此状态。
- 此任务可能会到期。此状态更改将使此任务结束。
- 您可以使用终止操作手动终止此任务。此状态更改将使此任务结束。

正常的任务流是，该服务返回输出消息或故障消息。如果返回了输出消息，那么此任务将进入“已完成”状态；如果返回了故障消息，那么此任务将进入“已失败”状态。在这两种情况下，该消息都可供任务发起者和启动者使用。

“已失败”、“已终止”、“已完成”和“已到期”状态都是结束状态。如果任务模板指定了自动删除，那么此任务将在删除计时器到期后被删除或者被手动删除。缺省情况下，调用任务不会被自动删除，因此被调用服务的结果可供访问。

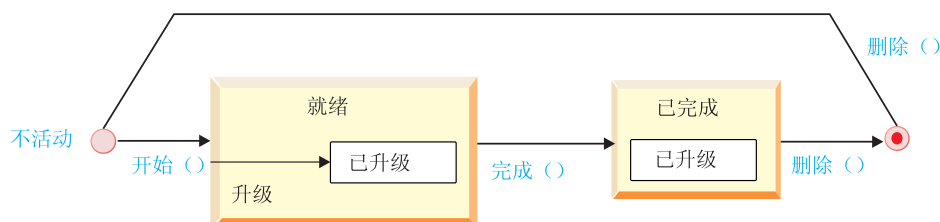
还有其他一些规则适用于直接插入调用任务。这些任务是业务流程的组成部分，因此该流程可以控制它们的生命周期：

- 如果业务流程通过业务流程管理器 API 或 SCA 客户机启动，那么创建该流程实例的活动的任务由该业务流程隐式地创建和启动。调用任务还可以由已运行的流程实例使用。在这种情况下，它们由流程创建，并且与接收或检取（接收选项）活动或者 on-event 事件处理程序相关联。
- 在业务流程中，此任务被表示为接收或检取（接收选项）活动或者 on-event 事件处理程序。如果对活动定义了直接插入调用任务，那么还将对此活动定义授权。
- 直接插入调用任务不具有“有效时间”，并且不能被手动终止。
- 如果此任务由业务流程隐式启动，那么它还将隐式地与业务流程一起被删除。
- 如果此任务由人员任务管理器 API 启动，那么它不会与流程一起被删除。如果设计此任务时指定了自动删除，那么它将在删除计时器到期后被删除。它还可以被手动删除。

管理任务

管理任务允许人员管理业务流程及其活动。如果没有可用的管理任务模板，那么在运行时，每当业务流程需要使用管理任务时，将创建缺省管理任务。

下图说明管理任务可能发生的状态过渡：



业务流程管理器在单一事务中隐式地创建和启动管理任务。因此，不活动状态在外部不可视，此任务将直接进入“就绪”状态。

“已完成”状态是一种结束状态。但是，此状态并不会阻止进一步的管理操作。

管理任务始终是直接插入任务，因此，它们的生命周期由业务流程控制。它们始终与业务流程一起被删除。

相关概念

第 39 页的『子任务』

当人员需要将分配给他们的部分工作委托他人完成，但希望继续对整体结果进行控制时，可以借助于子任务。子任务还可用于调用支持服务，以帮助人员完成所处理的任

第 36 页的『独立任务与直接插入任务』

面向服务的体系结构（SOA）模式提供了一组松散耦合组件的软件解决方案实现建议。遵循 SOA 模式的人员任务称为独立任务，而作为业务流程组成部分定义的人员任务称为直接插入任务。

任务调用方案

本节描述各种任务调用方法。

使用人员任务管理器 API 来调用任务组件

可以使用人员任务管理器 API 将任务实例化。人员任务管理器 API 客户端使用此 API 来创建和启动任务实例以及查询和处理任务实例。对于任务调用，此 API 提供了用于创建和启动下列各类任务的方法：

- 独立调用任务和直接插入调用任务
- 独立待执行任务
- 协作任务

由于管理任务必须在业务流程的上下文中进行调用，因此无法使用此 API 来调用这些任务。

此 API 支持下列任务交互样式：

- 以同步方式调用任务及相关服务

此交互样式使用 `callTask` 方法。对于单向操作而言，此调用在触发执行该任务和服务组件后返回。对于请求/响应操作而言，此调用将等待该服务和任务完成以及调用结果返回。

此交互样式只能应用于调用任务。

- 以异步方式调用任务及相关服务

此交互样式使用 `startTask` 方法。对于单向操作和请求/响应操作，此调用都将在触发执行该任务和服务组件后返回。并且，对于请求/响应操作而言，此调用以异步方式返回结果，此结果作为输出消息或故障消息存储在调用任务的上下文中。主调 API 客户机必须使用 API 方法通过程序检索此结果。此外，可以使用应答处理程序来确保当异步响应可用时尽快将其返回给客户机。

此调用样式可以应用于待执行任务、协作任务和调用任务。

人员任务管理器 API 作为 Enterprise JavaBeans (EJB) 实现和 Web Service 实现提供。这些实现的 API 方法都相似，但在功能范围方面有所不同。

有关这些 API 方法的更多信息，请参阅 `com.ibm.task.api` 包中 `HumanTaskManager` 接口的 Javadoc。

将待执行任务作为 SCA 服务组件进行调用

独立待执行任务代表可以由 SCA 客户机以异步方式调用的 SCA 服务组件。SCA 提供的机制可用于在 SCA 客户机与独立待执行任务之间进行连接。这包括用于定义下列各项的 SCA 方法：

- 连线，用于在 SCA 客户机（引用）与代表待执行任务的组件的接口之间进行连接
- 组件引用和接口的 SCA 限定符设置，用于控制交互样式、事务行为和交互可靠性等方面的特性

此外，独立待执行任务可以由作为业务流程实现的 SCA 客户机调用。在这种情况下，在 SCA 级和流程级都必须对连接加以考虑。从 SCA 级看来，SCA 客户机引用连接到 SCA 服务的接口。从流程级看来，调用活动的伙伴链接连接到待执行任务。

调用直接插入待执行任务

可以在长时间运行业务流程的人员任务活动的上下文中指定待执行任务。在这种情况下，在 SCA 级没有此任务的表示，此任务是代表该业务流程的 SCA 组件的组成部分。此任务充当人员任务活动的服务提供程序。在流程导航期间，每当遇到此活动时，将以异步方式调用这个待执行任务。

通过调用任务来调用 SCA 服务

独立调用任务充当相关 SCA 服务的访问组件。与此服务的关联是在 SCA 级定义的：此任务代表连接到 SCA 服务组件的 SCA 客户机。对调用任务进行的调用既涉及人员任务管理器级也涉及 SCA 级。调用任务本身由人员任务管理器 API 以同步方式或异步方式调用。然后，此任务（SCA 客户机）相应地调用相关联的 SCA 服务组件：如果此任务以同步方式被调用，那么以同步方式调用该组件，否则采用异步方式。

任务与服务之间的关联设计在 SCA 级进行。因此，SCA 提供的概念和机制可用于在独立调用任务与 SCA 服务组件之间进行连接。这包括用于定义下列各项的 SCA 方法：

- 连线，用于在 SCA 客户机引用与服务组件的接口之间进行连接
- 组件引用和接口的 SCA 限定符设置，用于控制交互样式、事务行为和交互可靠性等方面的特性

此外，独立调用任务可以连接到由业务流程实现的 SCA 组件。

通过直接插入调用任务来调用业务流程

可以在接收或检取活动的上下文中指定直接插入调用任务，也可以在业务流程的事件处理程序中进行指定。在 SCA 级没有此任务的表示，此任务是代表该业务流程的 SCA 组件的组成部分。但是，此任务充当该业务流程的客户机。每当人员任务管理器 API 调用此任务时，此任务将按照它的被调用方式来调用该业务流程。

为了管理业务流程和活动而创建的管理任务

管理任务允许人员管理业务流程及其活动。业务流程管理器为每个允许进行管理的业务流程或活动类型创建和启动管理任务。如果已对此流程或活动指定管理任务模板，那么将使用此模板。如果没有可用的模板，那么每当该业务流程需要使用管理任务时，都将创建缺省管理任务。

相关概念

『影响独立调用任务及其服务组件的行为的因素』

可以使用独立调用任务来运行与该任务的 SCA 组件相关联的服务组件体系结构（SCA）服务组件。该调用任务与该服务组件之间的关联是在 SCA 级设计的，即，通过将该任务组件的引用连接到相关服务组件的接口完成设计工作。许多因素会影响该调用任务及其相关服务组件的行为。

第 52 页的『方案：支持以异步方式调用服务的独立调用任务』

此方案仅考虑以异步方式调用任务和服务。此方案描述了服务组件体系结构（SCA）设置以及此类调用的预期事务行为和故障行为。

第 54 页的『方案：支持以异步方式和同步方式调用服务的独立调用任务』

此方案同时考虑以异步方式和同步方式调用任务及其相关服务。此方案描述了服务组件体系结构（SCA）设置以及这些类型的调用的预期事务行为和故障行为。

相关任务

第 493 页的『创建 API 事件处理程序』

当 API 方法处理人员任务时，将发生 API 事件。请使用 API 事件处理程序插件服务供应商接口（SPI）来创建插件，以处理 API 发送的任务事件或具有相同 API 事件的内部事件。

影响独立调用任务及其服务组件的行为的因素

可以使用独立调用任务来运行与该任务的 SCA 组件相关联的服务组件体系结构（SCA）服务组件。该调用任务与该服务组件之间的关联是在 SCA 级设计的，即，通过将该任务组件的引用连接到相关服务组件的接口完成设计工作。许多因素会影响该调用任务及其相关服务组件的行为。

WSDL 操作类型

SCA 引用和 SCA 接口与包含一个或多个操作的 WSDL 端口类型相关联。每个操作都可以是单向操作或请求/响应操作：

- 单向操作是指主调任务不可知其是否已完成的服务执行。此任务服务执行以成功调用相关服务作为结束。
- 请求/响应操作是指主调任务可知其是否已完成的服务执行。当主调任务获得服务执行结果时，此任务执行结束。

API 调用方法

人员任务管理器 API 支持下列任务交互样式:

- 使用 callTask 方法，以同步方式调用任务及相关服务
- 使用 startTask 方法，以异步方式调用任务及相关服务

服务组件的执行持续时间

设置执行持续时间值时必须考虑系统上其他工作负载引起的预期开销。此外，还必须结合您对业务流程编排器所在服务器设置的事务超时值对执行持续时间加以考虑。在您决定允许以同步方式调用带有请求/响应接口的服务组件前，请对这些值进行比较。在此类情况下，服务组件的执行时间必须小于对服务器设置的事务超时值。

SCA 限定符设置

任务组件引用和服务组件接口只允许使用 SCA 限定符的某些组合。

相关概念

第 49 页的『任务调用方案』
本节描述各种任务调用方法。

方案: 支持以异步方式调用服务的独立调用任务

此方案仅考虑以异步方式调用任务和服务。此方案描述了服务组件体系结构 (SCA) 设置以及此类调用的预期事务行为和故障行为。

此方案适用于仅使用异步调用的人员任务管理器 API 客户机，例如业务流程编排器资源管理器。采用此方案时，不需要在设计任务时评估与该任务相关联的服务的执行持续时间。

任务组件设置

任务组件可以具有下列设置。如果使用 WebSphere Integration Developer 来定义任务组件，那么将自动生成 type 属性的有效值。

限定符类型: type 属性	值
引用属性: Multiplicity	1:1 (必需)
引用限定符: DeliverAsyncAt	commit (必需)
实现限定符*: Transaction	global (必需)
引用限定符**: SuspendTransaction	不适用
实现限定符***: ActivitySession	true (必需)
引用限定符***: SuspendActivitySession	false (缺省)
引用限定符: Reliability	assured (必需)
引用限定符: RequestExpiration	any
引用限定符: ResponseExpiration	any

限定符类型: type 属性	值
注: <ul style="list-style-type: none"> • *: 如果使用事务设置, 那么请使用 <code>global</code>; 如果使用活动会话设置, 那么请使用 <code>local</code>。 • **: 如果事务设置为 <code>global</code>, 那么仅使用事务设置 • ***: 如果事务设置为 <code>local</code>, 那么仅使用活动会话设置 	

服务组件设置

服务组件可以具有下列设置。如果使用 WebSphere Integration Developer 来定义任务组件, 那么将自动生成 `type` 属性的有效值。

限定符类型: type 属性	值
接口属性: <code>PreferredInteractionStyle</code>	忽略
实现限定符*: <code>Transaction</code>	<code>local</code> (缺省) <code>global</code>
接口限定符**: <code>JoinTransaction</code>	<code>false</code> (缺省) <code>true</code>
实现限定符***: <code>ActivitySession</code>	<code>any</code> (缺省)
接口限定符***: <code>JoinActivitySession</code>	<code>false</code> (缺省)
注: <ul style="list-style-type: none"> • *: 如果使用事务设置, 那么请使用 <code>global</code>; 如果使用活动会话设置, 那么请使用 <code>local</code>。 • **: 如果事务设置为 <code>global</code>, 那么仅使用事务设置 • ***: 如果事务设置为 <code>local</code>, 那么仅使用活动会话设置 	

以下列表提供了服务的 **Transaction** 和 **JoinTransaction** 限定符设置的有效组合:

- **Transaction** 限定符设置为 `local`, **JoinTransaction** 设置为 `false`。使用这些设置时, 任务与服务调用在不同的事务中运行。
- **Transaction** 限定符设置为 `global`, **JoinTransaction** 设置为 `false`。使用这些设置时, 任务与服务调用在不同的事务中运行。
- **Transaction** 限定符设置为 `global`, **JoinTransaction** 设置为 `true`。使用这些设置时, 任务与服务调用在同一个事务中运行。

事务行为和故障行为

在此异步调用方案中, `startTask` 方法仅用于 API 调用。任务与服务调用在不同的事务中执行。发生服务实现未处理的运行时异常时, 下列内容适用。此方案采用下列事务行为和异常处理。

操作类型	何时发生 SCA 运行时异常	任务和服务的行为
单向操作	在服务调用期间, 但在开始执行该服务之前	该任务接收到 SCA 运行时异常。人员任务管理器 API 方法抛出 <code>CoreOTaskServiceRuntimeExceptionReceivedException</code> 异常。任务事务回滚, 该任务保持处于不活动状态。
单向操作	在服务执行期间	调用任务不会接收到通知。该任务将进入“已完成”状态。将生成可以由失败事件管理器处理的失败事件。

操作类型	何时发生 SCA 运行时异常	任务和服务的行为
请求/响应操作	在服务调用期间，但在开始执行该服务之前	该任务接收到 SCA 运行时异常。人员任务管理器 API 方法抛出 CoreOTaskServiceRuntimeExceptionReceivedException 异常。任务事务回滚，该任务保持处于不活动状态。
请求/响应操作	在服务执行期间	该任务接收到 SCA 运行时异常通知并将其存储在任务上下文中的数据库中。如果有可用的应答处理程序，那么将使用该处理程序来通知该客户机。该任务将进入“已失败”状态。

操作定义可以包含一条或多条可能由服务组件在执行期间抛出的故障消息。

任务组件将接收到有关故障消息的通知，如下所示：

- 将该故障消息存储在任务上下文中的数据库中
- 该任务进入“已失败”状态
- 如果该任务以同步方式被调用，并且已指定应答处理程序，那么将调用该应答处理程序以便将该故障返回给客户机
- 如果该任务以异步方式被调用，那么该故障消息将作为 FaultReplyException 异常返回给客户机

故障处理不影响事务行为。事务不会回滚。

相关概念

第 49 页的『任务调用方案』

本节描述各种任务调用方法。

方案：支持以异步方式和同步方式调用服务的独立调用任务

此方案同时考虑以异步方式和同步方式调用任务及其相关服务。此方案描述了服务组件体系结构（SCA）设置以及这些类型的调用的预期事务行为和故障行为。

在此方案中，人员任务管理器客户机既执行异步调用也执行同步调用。这意味着，您已评估服务执行时间是否小于预期的服务器事务超时值。通常，执行持续时间必须充分小于服务器事务超时值。

任务组件设置

任务组件可以具有下列设置。如果使用 WebSphere Integration Developer 来定义任务组件，那么将自动生成 type 属性的有效值。

限定符类型: type 属性	值
引用属性: Multiplicity	1:1 (必需)
引用限定符: DeliverAsyncAt	commit (必需)
实现限定符*: Transaction	global (必需)
引用限定符**: SuspendTransaction	不适用
实现限定符***: ActivitySession	true (必需)
引用限定符***: SuspendActivitySession	false (缺省)
引用限定符: Reliability	assured (必需)
引用限定符: RequestExpiration	any

限定符类型: type 属性	值
引用限定符: ResponseExpiration	any
注:	
<ul style="list-style-type: none"> *: 如果使用事务设置, 那么请使用 global; 如果使用活动会话设置, 那么请使用 local。 ** : 如果事务设置为 global, 那么仅使用事务设置 ***: 如果事务设置为 local, 那么仅使用活动会话设置 	

服务组件设置

服务组件可以具有下列设置。如果使用 WebSphere Integration Developer 来定义任务组件, 那么将自动生成 type 属性的有效值。

限定符类型: type 属性	值
接口属性: PreferredInteractionStyle	忽略
实现限定符*: Transaction	local (缺省) global
接口限定符**: JoinTransaction	false (缺省) true
实现限定符***: ActivitySession	any (缺省)
接口限定符***: JoinActivitySession	false (缺省)
注:	
<ul style="list-style-type: none"> *: 如果使用事务设置, 那么请使用 global; 如果使用活动会话设置, 那么请使用 local。 ** : 如果事务设置为 global, 那么仅使用事务设置 ***: 如果事务设置为 local, 那么仅使用活动会话设置 	

以下列表提供了服务的 **Transaction** 和 **JoinTransaction** 限定符设置的有效组合:

- **Transaction** 限定符设置为 local, **JoinTransaction** 设置为 false。使用这些设置时, 任务与服务调用在不同的事务中运行。
- **Transaction** 限定符设置为 global, **JoinTransaction** 设置为 false。使用这些设置时, 任务与服务调用在不同的事务中运行。
- **Transaction** 限定符设置为 global, **JoinTransaction** 设置为 true。使用这些设置时, 任务与服务调用在同一个事务中运行。

事务行为和故障行为

此方案采用下列事务行为和异常处理。

API 调用样式	操作类型	何时发生 SCA 运行时异常	任务和服务的行为
callTask	单向操作	在服务调用期间, 但在开始执行该服务之前	该任务接收到 SCA 运行时异常。人员任务管理器 API 方法抛出 CoreOTaskServiceRuntimeExceptionReceivedException 异常。任务事务回滚, 该任务保持处于不活动状态。
callTask	单向操作	在服务执行期间	该任务接收到 SCA 运行时异常。人员任务管理器 API 方法抛出 CoreOTaskServiceRuntimeExceptionReceivedException 异常。任务事务回滚, 该任务保持处于不活动状态。

API 调用样式	操作类型	何时发生 SCA 运行时异常	任务和服务的行为
callTask	请求/响应操作	在服务调用期间，但在开始执行该服务之前	该任务接收到 SCA 运行时异常。人员任务管理器 API 方法抛出 CoreOTaskServiceRuntimeExceptionReceivedException 异常。任务事务回滚，该任务保持处于不活动状态。
callTask	请求/响应操作	在服务执行期间	该任务接收到 SCA 运行时异常。人员任务管理器 API 方法抛出 CoreOTaskServiceRuntimeExceptionReceivedException 异常。任务事务回滚，该任务保持处于不活动状态。
startTask	单向操作	在服务调用期间，但在开始执行该服务之前	该任务接收到 SCA 运行时异常。人员任务管理器 API 方法抛出 CoreOTaskServiceRuntimeExceptionReceivedException 异常。任务事务回滚，该任务保持处于不活动状态。
startTask	单向操作	在服务执行期间	调用任务不会接收到通知。该任务将进入“已完成”状态。将生成可以由失败事件管理器处理的失败事件。
startTask	请求/响应操作	在服务调用期间，但在开始执行该服务之前	该任务接收到 SCA 运行时异常。人员任务管理器 API 方法抛出 CoreOTaskServiceRuntimeExceptionReceivedException 异常。任务事务回滚，该任务保持处于不活动状态。
startTask	请求/响应操作	在服务执行期间	该任务接收到 SCA 运行时异常通知并将其存储在任务上下文中的数据库中。如果有可用的应答处理程序，那么将使用该处理程序来通知该客户机。该任务将进入“已失败”状态。

操作定义可以包含一条或多条可能由服务组件在执行期间抛出的故障消息。

任务组件将接收到有关故障消息的通知，如下所示：

- 将该故障消息存储在任务上下文中的数据库中
- 该任务进入“已失败”状态
- 如果该任务以异步方式被调用，并且已指定应答处理程序，那么将调用该应答处理程序以便将该故障返回给客户机
- 如果该任务以同步方式被调用，那么该故障消息将作为 FaultReplyException 异常返回给客户机

故障处理不影响事务行为。事务不会回滚。

相关概念

第 49 页的『任务调用方案』

本节描述各种任务调用方法。

授权与人员分配

授权是使某些人员能够对任务模板、任务实例和升级执行所选操作的机制。授权角色用于定义可供特定角色使用的一组操作。可以使用 J2EE 机制将人员分配给系统级角色，也可以使用人员分配条件将人员分配给任务实例角色。

人员任务的授权角色

可以对人员任务执行的操作取决于您的授权角色。此角色可以是系统级 J2EE 角色，也可以是基于实例的角色。

系统级 Java 2 Platform, Enterprise Edition (J2EE) 角色是在人员任务管理器配置期间设置的。这些角色所代表的权限级别对所有任务和升级有效。基于实例的角色对各个

任务和升级实例有效，或者对用于创建任务或升级实例的模板有效。基于角色的授权要求对应用程序服务器启用管理安全性和应用程序安全性。

J2EE 角色

支持下列 J2EE 角色：

- **TaskSystemAdministrator**。分配到此角色的用户拥有所有特权。此角色也称为“人员任务的系统管理员”。
- **TaskSystemMonitor**。分配到此角色的用户能够查看所有任务对象的属性。此角色也称为“人员任务的系统监视员”。

您可以使用管理控制台来更改对这些角色分配的用户和组。

基于实例的角色

任务实例或升级实例不会被直接分配给人员，而是与人员被分配到的预定义角色相关联。分配给基于实例的角色的任何人员都能够执行该角色的操作。用户与基于角色的实例之间的关联由人员分配确定或者作为任务操作结果确定。

在运行时，人员分配工作根据人员目录中存储的用户和用户组信息将人员分配给下列角色：潜在创建者、潜在启动者、潜在所有者、阅读者、编辑者、管理员和升级接收者。下列角色仅与一个用户相关联，并且作为任务操作结果进行分配：发起者、启动者和所有者。

这些角色有权执行下列操作：

角色	有权执行的操作
潜在创建者	此角色的成员能够创建任务实例。如果没有为任务模板定义潜在实例创建者，那么将把所有用户都视为此角色的成员。
发起者	此角色的人员在任务启动前拥有管理权。任务启动后，发起者将拥有阅读者的权限，并且能够执行一些管理操作，例如暂挂和恢复任务以及移交工作项。
潜在启动者	此角色的成员能够启动现有的任务实例。如果未指定潜在启动者，那么发起者将成为潜在启动者。对于没有潜在启动者的直接插入任务而言，缺省值是“每个人”。
启动者	此角色的人员拥有阅读者的权限，并且能够执行一些管理操作，例如移交工作项。
潜在所有者	此角色的成员能够声明任务。如果潜在所有者不是针对任务模板定义的，那么将把所有用户都视为此角色的成员。如果对此角色执行的人员解析操作失败，那么将把管理员指定为潜在所有者。
所有者	此角色的人员能够处理并完成任务。
阅读者	此角色的成员能够查看所有任务对象的属性，但无权对它们进行处理。
编辑者	此角色的成员能够处理任务内容，但无权声明或完成任务。
管理员	此角色的成员能够管理任务、任务模板和升级。
升级接收者	此角色的成员拥有升级阅读者和已升级任务阅读者的权限。

相关概念

第 28 页的『业务流程的授权角色』

角色是授权级别相同的一组人员。可以对业务流程执行的操作取决于您的授权角色。此角色可以是 J2EE 角色，也可以是基于实例的角色。

第 39 页的『子任务』

当人员需要将分配给他们的部分工作委托他人完成，但希望继续对整体结果进行控制时，可以借助于子任务。子任务还可用于调用支持服务，以帮助人员完成所处理的任务。

第 74 页的『缺省人员分配』

如果未定义某些任务角色的人员分配条件，或者人员解析失败或未返回结果，那么将以缺省方式进行人员分配。直接插入任务和独立任务的缺省人员分配并不相同。

任务类型的基于实例的授权角色

基于实例的授权角色在任务建模时与人员任务和升级相关联。任务类型确定了特定授权角色是否可用。

角色	待执行任务	调用任务	协作任务	管理任务	注释
潜在实例创建者	X	X	X		能够创建任务实例的人员
发起者	X	X	X		创建任务的人员
潜在所有者	X		X		能够声明和处理任务的人员
所有者	X		X		声明任务的人员
潜在启动者		X			能够启动任务的人员
启动者		X			启动任务的人员
管理员	X	X	X	X ¹	能够管理任务的人员
编辑者	X		X		能够编辑任务数据的人员
阅读者	X	X	X	X ²	能够查看任务数据的人员
升级接收者	X ³	X ³	X ³	X ³	接收升级的人员
注: 1. 此角色还有权对受管流程或活动执行管理操作 2. 此角色还有权对受管流程或活动执行读取操作 3. 此角色有权对根据这些任务类型生成的升级执行操作，但无权对任务本身执行操作					

授权与工作项

每个任务角色都使用户能够对相关任务执行一组具体的操作。人员的授权通过工作项进行管理。工作项代表所分配人员与该任务角色所代表任务操作之间的关系。

工作项具有下列各项内容:

- 用户或用户组的身份
- 操作可以处理的任务的身份
- 与用户相关联的任务角色

可以通过下列其中一种方法指定与工作项相关联的人员:

- 作为一个确切的标识。这将生成用户工作项。
- 作为一个确切的标识。这将生成组工作项。

- 通过使用 **Everybody** 人员分配条件，对每个用户进行指定。这将生成 Everybody 工作项。

业务流程编排器的授权机制确保当下列其中一个条件成立时，用户能够执行与工作项相关联的操作：

- 该用户登录时，使用的用户标识与该用户工作项的所指定用户标识匹配
- 登录用户隶属于组工作项的所指定组标识的对应组
- 该工作项被分配给每个人

人员任务管理器 API 提供了用于查询人员任务、升级和其他对象的方法。运行查询时，通过仅返回用户所拥有的工作项的相应数据，确保用户有权查看所查询的数据。还可以使用 API 来管理基于实例的授权操作。此功能通过创建和删除工作项以及通过在人员之间移交工作项实现。有关这些 API 方法的更多信息，请参阅 `com.ibm.task.api` 包中 `HumanTaskManager` 接口的 Javadoc。

人员分配条件

人员分配条件是一些构造，任务模型使用这些构造来标识多组可以被赋予基于实例的授权角色的人员。在运行时，人员解析功能使用人员分配条件来检索用户标识和其他信息，以便完成撰写电子邮件之类的工作。在运行时，当通过程序创建任务模型时，也将使用人员分配条件。

在 WebSphere Integration Developer 中，可以使用人员分配条件定义（以前称为“人员动词”）来设计任务角色的人员分配情况。此定义包含查询名称和一组查询参数。部署任务时，分配条件将变换为特定于人员目录（例如虚拟成员管理器）的查询。任务运行时，这些查询将检索一组已被赋予角色（例如潜在所有者）的人员。

以下示例说明实现任务角色的人员分配条件定义时所需执行的步骤：

1. 在 WebSphere Integration Developer 中，设计者使新任务与人员目录配置（例如虚拟成员管理器 `bpc/staff/samplevmmconfiguration`）相关联。

此步骤确定可用于进行人员分配的人员分配条件。

2. 在 WebSphere Integration Developer 中，设计者使任务角色与人员分配条件定义相关联。

例如，潜在所有者角色与人员分配条件 **Group Members** 相关联并设置下列参数：

- 将 **GroupName** 设置为值 `cn=group1, dc=mycomp, dc=com`
 - 将 **IncludeSubgroups** 设置为值 `true`
3. 部署任务时，人员分配服务将确定所要使用的人员目录提供程序。它将人员分配条件变换为用于该人员目录提供程序的查询，此查询将存储在内部。

根据所使用人员目录的不同，设计任务时可以使用预定义人员分配条件的不同子集：

- LDAP 和虚拟成员管理器人员目录提供程序支持全部预定义的人员分配条件定义
- 用户注册表人员目录提供程序只支持那些基于用户名和组名的定义。不支持基于经理和电子邮件属性的定义。
- 系统人员目录提供程序仅用于测试用途。由于只支持指定一组硬编码的用户标识，因此不需要访问人员目录。

预定义人员分配条件

系统提供了预定义的人员分配条件，用于从人员目录中检索一组用户。

在 WebSphere Integration Developer 中，可以使用人员分配条件（以前称为“人员动词”）来指定人员任务中的人员分配情况。在建模和部署期间，这些条件将变换为一组可以对人员目录运行的查询。本主题列示下列预定义人员分配条件的参数：

- Department Members
- Everybody
- Group
- Group Members
- Group Members without Named Users
- Group Members without Filtered Users
- Group Search
- Manager of Employee
- Manager of Employee by user ID
- Native Query
- Nobody
- Person Search
- Role Members
- User Records by user ID
- User Records by user ID without Named Users
- Users
- Users by user ID
- Users by user ID without Named Users

在指定人员分配条件时，请考虑下列事项：

- 如果您正在处理大批人员，那么最好使用 **Group** 人员分配条件，这是因为此条件将全部组成员作为一个整体进行处理。这样，您就可以轻松方便地将一个组的人员任务移交给另一个组。人员的组成员关系将在该人员登录并访问人员任务时进行解析。
- 如果要对人员任务逐个分配属于某个组的人员，那么可以使用 **Group Members** 人员分配条件代替进行组分配。**Group Members** 人员分配条件为每位人员逐个创建分配项。然后，可以将此分配项移交给另一位人员。可以进行替代，即，当某位人员不在场时，他可以被另一位人员替换。**Group Members without Named Users** 是此人员分配条件的一种变体，它支持“划分职责”分配模式。

注： 逐个地将人员分配到一个组会对运行时性能产生负面影响，在将众多人员分配到该组时尤其如此。

- 要将几个不属于同一个组的人员分配到某个人员任务，请考虑使用 **User Records by user ID** 人员分配条件定义。如果设计期间未以静态方式定义人员分配，而是让人员分配包含替换表达式，那么也可以使用此定义。替换表达式可以引用定制属性或者人员任务输入消息。**Users by user ID** 人员分配条件定义与 **User Records by user ID** 定义类似。虽然 **Users by user ID** 定义在运行时的性能优于 **User Records by user ID** 定义，但提供的功能较少：
 - 它不检查输入的用户标识是否正确
 - 它不检索所指定用户标识的电子邮件地址之类的信息，这导致它不适用于将人员分配给电子邮件升级

- Everybody 人员分配条件定义也值得考虑。此定义表示将所有已认证的用户分配给人员任务。尽管在某些情况下，公司内所有人员都可以执行特定的工作，但此定义在开发期间以及快速设计应用程序原型期间特别有用。

Department Members

使用此条件来检索部门成员。此条件受轻量级目录访问协议（LDAP）和虚拟成员管理器人员目录提供程序支持。

参数	用法	类型	描述
DepartmentName	必需	字符串	要检索的用户的部门名。部门名必须具有下列其中一个值： <ul style="list-style-type: none"> • 对于虚拟成员管理器：虚拟成员管理器组的唯一名称 • 对于 LDAP：LDAP 组的专有名称（DN）
IncludeNestedDepartments	必需	布尔值	指定在查询中是否处理嵌套的部门。
AlternativeDepartmentName1	可选	字符串	用户所属的附加部门。
AlternativeDepartmentName2	可选	字符串	用户所属的附加部门。

Everybody

使用此条件将 WebSphere Process Server 所认证的每个用户分配到任务角色。此条件没有参数。

此条件受所有人员目录提供程序支持。

Group

使用此条件将一个组分配到任务角色。此分配将创建组工作项，而不会为每个所分配的用户创建用户工作项。

此条件受所有人员目录提供程序支持。

参数	用法	类型	描述
GroupId	必需	字符串	要检索的用户的组名。此参数支持替换表达式。组标识必须具有下列其中一个值： <ul style="list-style-type: none"> • 对于虚拟成员管理器：组条目的唯一名称 • 对于 LDAP：组条目的 DN • 对于用户注册表提供程序，使用的名称格式取决于对任务部署所在应用程序服务器设置的用户存储库： <ul style="list-style-type: none"> – 对于本地操作系统，请使用本地操作系统支持的组名 – 对于独立定制注册表，请使用该定制实现支持的组名 – 对于独立 LDAP 注册表，请使用组条目的 DN

Group Members

使用此条件来检索组成员。此条件受 LDAP、虚拟成员管理器和用户注册表人员目录提供程序支持。

参数	用法	类型	描述
GroupName	必需	字符串	要检索的用户的组名。此参数支持替换表达式。组标识必须具有下列其中一个值： <ul style="list-style-type: none">• 对于虚拟成员管理器：组条目的唯一名称• 对于 LDAP：组条目的 DN• 对于用户注册表提供程序，使用的名称格式取决于对任务部署所在应用程序服务器设置的用户存储库：<ul style="list-style-type: none">– 对于本地操作系统，请使用本地操作系统支持的组名– 对于独立定制注册表，请使用该定制实现支持的组名– 对于独立 LDAP 注册表，请使用组条目的 DN
IncludeSubgroups	必需	布尔值	指定在查询中是否处理嵌套的子组。
AlternativeGroupName1	可选	字符串	用户所属的其他组。
AlternativeGroupName2	可选	字符串	用户所属的其他组。

Group Members without Named Users

使用此条件来检索一个组中除明确指定的用户以外的所有成员。此条件受 LDAP、虚拟成员管理器和用户注册表人员目录提供程序支持。

参数	用法	类型	描述
GroupName	必需	字符串	要检索的用户的组名。此参数支持替换表达式。组标识必须具有下列其中一个值： <ul style="list-style-type: none">• 对于虚拟成员管理器：组条目的唯一名称• 对于 LDAP：组条目的 DN• 对于用户注册表提供程序，使用的名称格式取决于对任务部署所在应用程序服务器设置的用户存储库：<ul style="list-style-type: none">– 对于本地操作系统，请使用本地操作系统支持的组名– 对于独立定制注册表，请使用该定制实现支持的组名– 对于独立 LDAP 注册表，请使用组条目的 DN
IncludeSubgroups	必需	布尔值	指定在查询中是否处理嵌套的子组。
NamedUsers	必需	字符串	要从所检索组成员列表中排除的用户的用户标识。此参数支持替换表达式。

Group Members without Filtered Users

使用此条件来检索一个组中除搜索过滤器所定义的一组用户以外的所有成员。此条件受 LDAP 和虚拟成员管理器人员目录提供程序支持。

参数	用法	类型	描述
GroupName	必需	字符串	要检索的用户的组名。此参数支持替换表达式。组标识必须具有下列其中一个值： <ul style="list-style-type: none">• 对于虚拟成员管理器：组条目的唯一名称• 对于 LDAP：组条目的 DN
IncludeSubgroups	必需	布尔值	指定在查询中是否处理嵌套的子组。
FilterAttribute	必需	字符串	在搜索过滤器中使用的属性的名称。
FilterValue	必需	字符串	在搜索过滤器中使用的过滤器值。可以在该过滤器中使用星号 (*) 通配符。

Group Search

使用此条件来根据匹配的属性搜索组以及指定该组的成员。此条件受 LDAP 和虚拟成员管理器人员目录提供程序支持。

参数	用法	类型	描述
GroupID	可选	字符串	要检索的用户的组标识。
Type	可选	字符串	要检索的用户的组类型。
IndustryType	可选	字符串	用户所属的组的行业类型。
BusinessType	可选	字符串	用户所属的组的业务类型。
GeographicLocation	可选	字符串	用户所在位置的指示。
Affiliates	可选	字符串	用户的雇员。
DisplayName	可选	字符串	组的显示名。
Secretary	可选	字符串	用户的秘书。
Assistant	可选	字符串	用户的助手。
Manager	可选	字符串	用户的经理。
BusinessCategory	可选	字符串	用户所属的组的业务类别。
ParentCompany	可选	字符串	用户的母公司。

对于虚拟成员管理器，Group 实体具有与下列 Group Search 条件参数等同的属性：

- GS_GroupID: cn
- GS_DisplayName: displayName
- GS_BusinessCategory: businessCategory

Manager of Employee

使用此条件来按人员姓名检索该人员的经理。此条件受 LDAP 和虚拟成员管理器人员目录提供程序支持。

参数	用法	类型	描述
EmployeeName	必需	字符串	职员姓名，将检索此职员的经理。职员姓名必须具有下列其中一个值： <ul style="list-style-type: none"> • 对于虚拟成员管理器：人员条目的唯一名称 • 对于 LDAP：人员条目的 DN

Manager of Employee by user ID

使用此条件来按人员的用户标识检索该人员的经理。此条件受 LDAP 和虚拟成员管理器人员目录提供程序支持。

参数	用法	类型	描述
EmployeeUserID	必需	字符串	职员的登录用户标识，将检索此职员的经理。此参数支持替换表达式。

Native Query

使用此条件来根据特定于目录的参数定义本机查询。此条件受 LDAP 和虚拟成员管理器人员目录提供程序支持。

参数	用法	类型	描述
QueryTemplate	必需	字符串	要使用的查询模板。此参数必须具有下列其中一个值： search、user 和 usersOfGroup。
Query	必需	字符串	指定查询。此参数支持替换表达式。查询类型取决于查询模板。 <ul style="list-style-type: none"> • search 模板：搜索过滤器 <ul style="list-style-type: none"> – 对于虚拟成员管理器：有效的搜索表达式 – 对于 LDAP：有效的 LDAP 过滤器 • user 模板：用户 DN <ul style="list-style-type: none"> – 对于虚拟成员管理器：用户条目的唯一名称 – 对于 LDAP：用户条目的 DN • usersOfGroup：组 DN <ul style="list-style-type: none"> – 对于虚拟成员管理器：组的唯一名称 – 对于 LDAP：组的 DN
AdditionalParameter1	可选	字符串	指定查询。此参数支持替换表达式。参数类型取决于查询模板。 <ul style="list-style-type: none"> • search 模板。用于指定是否执行递归搜索。支持的值是： yes 和 no。 • user 模板。不受支持。 • usersOfGroup。用于指定是否执行递归搜索。支持的值是： yes 和 no。
AdditionalParameter2	可选	字符串	使用此条件来指定要搜索的基本条目。 <ul style="list-style-type: none"> • 对于虚拟成员管理器：基本条目的唯一名称，例如 dc=mycomp, dc=com • 对于 LDAP：基本条目的 DN

参数	用法	类型	描述
AdditionalParameter3	可选	字符串	使用此条件来指定附加参数。 如果使用了缺省的映射 XSLT 文件，那么不支持此参数。
AdditionalParameter4	可选	字符串	使用此条件来指定附加参数。 如果使用了缺省的映射 XSLT 文件，那么不支持此参数。
AdditionalParameter5	可选	字符串	使用此条件来指定附加参数。 如果使用了缺省的映射 XSLT 文件，那么不支持此参数。

Nobody

使用此条件来拒绝用户对任务角色的访问权。只有所继承的授权以及人员解析缺省值随此条件应用。此条件没有参数。

Person Search

使用此条件来根据匹配的属性搜索人员。此条件受 LDAP、虚拟成员管理器和用户注册表人员目录提供程序支持。

参数	用法	类型	描述
UserID	可选	字符串	要检索的用户的用户标识。
Profile	可选	字符串	要检索的用户的概要文件。
LastName	可选	字符串	要检索的用户的姓氏。
FirstName	可选	字符串	要检索的用户的名字。
MiddleName	可选	字符串	要检索的用户的中间名。
Email	可选	字符串	用户的电子邮件地址。
Company	可选	字符串	用户所属的公司。
DisplayName	可选	字符串	用户的显示名。
Secretary	可选	字符串	用户的秘书。
Assistant	可选	字符串	用户的助手。
Manager	可选	字符串	用户的经理。
Department	可选	字符串	用户所属的部门。
Phone	可选	字符串	用户的电话号码。
Fax	可选	字符串	用户的传真号码。
Gender	可选	字符串	用户的性别。
Timezone	可选	字符串	用户所在的时区。
PreferredLanguage	可选	字符串	用户的首选语言。

对于虚拟成员管理器，PersonAccount 实体具有与下列 People Search 条件参数等同的属性：

- PS_UserID: uid

- PS_LastName: sn
- PS_FirstName: givenName
- PS_MiddleName: initials
- PS_Email: mail
- PS_DisplayName: displayName
- PS_Secretary: secretary
- PS_Manager: manager
- PS_Department: departmentNumber
- PS_Phone: telephoneNumber
- PS_PREFERREDLanguage: preferredLanguage

Role Members

使用此条件来检索与角色相关联的用户。此条件受 LDAP 和虚拟成员管理器人员目录提供程序支持。

参数	用法	类型	描述
RoleName	必需	字符串	要检索的用户的角色名。
IncludeNestedRoles	必需	布尔值	指定在查询中是否处理嵌套的角色。
AlternativeRoleName1	可选	字符串	用户的附加角色名。
AlternativeRoleName2	可选	字符串	用户的附加角色名。

User Records by User ID

使用此条件为具有已知用户标识的用户定义查询。此条件受 LDAP 和虚拟成员管理器人员目录提供程序支持。此条件返回这些用户的用户标识、电子邮件信息以及已设置的首选语言环境。

参数	用法	类型	描述
UserID	必需	字符串	要检索的用户的用户标识。此参数支持替换表达式。
AlternativeID1	可选	字符串	附加的用户标识。使用此参数来检索多个用户。
AlternativeID2	可选	字符串	附加的用户标识。使用此参数来检索多个用户。

Users Records by User ID without Named Users

使用此条件为具有已知用户标识的用户定义查询并排除明确指定的用户标识。此条件受 LDAP、虚拟成员管理器和用户注册表人员目录提供程序支持。此条件返回这些用户的用户标识和电子邮件信息。

参数	用法	类型	描述
UserID	必需	字符串	要检索的用户的用户标识。此参数支持替换表达式。
AlternativeID1	可选	字符串	附加的用户标识。使用此参数来检索多个用户。
AlternativeID2	可选	字符串	附加的用户标识。使用此参数来检索多个用户。

参数	用法	类型	描述
NamedUsers	必需	字符串	要从用户标识列表中排除的用户的用户标识。此参数支持替换表达式。

Users

使用此条件来定义通过名称标识的用户的查询。此条件受所有人员目录提供程序支持。

参数	用法	类型	描述
Name	必需	字符串	要检索的用户的名称。 <ul style="list-style-type: none"> 对于虚拟成员管理器：人员条目的唯一名称 对于 LDAP：人员条目的 DN 对于用户注册表提供程序，使用的名称格式取决于对任务部署所在应用程序服务器设置的用户存储库： <ul style="list-style-type: none"> 对于本地操作系统，请使用要分配的用户的用户标识 对于独立定制注册表，请使用该定制实现支持的人员名 对于独立 LDAP 注册表，请使用人员条目的 DN
AlternativeName1	可选	字符串	附加的用户名。使用此参数来检索多个用户。
AlternativeName2	可选	字符串	附加的用户名。使用此参数来检索多个用户。

Users by User ID

使用此条件为具有已知用户标识的用户定义查询。使用短名称来指定值，例如 wpsadmin。此条件不要求访问人员目录。此条件受所有人员目录提供程序支持。

参数	用法	类型	描述
UserID	必需	字符串	要检索的用户的用户标识。此参数支持替换表达式。
AlternativeID1	可选	字符串	附加的用户标识。使用此参数来检索多个用户。
AlternativeID2	可选	字符串	附加的用户标识。使用此参数来检索多个用户。

Users by User ID without Named Users

使用此条件为具有已知用户标识的用户定义查询并排除明确指定的用户标识。使用短名称来指定值，例如 wpsadmin。此条件不要求访问人员存储库。此条件受所有人员目录提供程序支持。

参数	用法	类型	描述
UserID	必需	字符串	要检索的用户的用户标识。此参数支持替换表达式。
AlternativeID1	可选	字符串	附加的用户标识。使用此参数来检索多个用户。

参数	用法	类型	描述
AlternativeID2	可选	字符串	附加的用户标识。使用此参数来检索多个用户。
NamedUsers	必需	字符串	要从用户标识列表中排除的用户的用户标识。此参数支持替换表达式。

相关概念

『人员分配条件定义中的替换表达式』

在某些人员分配条件定义中，可以使用替换表达式作为参数值。在运行时，人员解析功能可以根据上下文提供的信息来解析分配条件。

人员分配条件定义中的替换表达式

在某些人员分配条件定义中，可以使用替换表达式作为参数值。在运行时，人员解析功能可以根据上下文提供的信息来解析分配条件。

例如，以下人员分配条件定义指定了 `htm:input.\name` 替换表达式作为参数：

```
<verb>
<name>Users by user ID</name>
  <parameter id="UserID">%htm:input.\name%</parameter>
</verb>
```

此变量表示该任务在启动时接收的任务输入消息值的“name”元素。人员解析功能将动态地将该表达式替换为实际的任务输入消息值。

相关信息

第 60 页的『预定义人员分配条件』

系统提供了预定义的人员分配条件，用于从人员目录中检索一组用户。

人员解析

人员解析根据一组参数化查询表达式（称为人员分配条件）从人员目录中检索用户信息。

相关概念

第 28 页的『业务流程的授权角色』

角色是授权级别相同的一组人员。可以对业务流程执行的操作取决于您的授权角色。此角色可以是 J2EE 角色，也可以是基于实例的角色。

人员目录

人员目录存储用于进行人员解析的用户信息。

为了支持人员解析，人员目录必须支持下列属性：

- 用于标识用户概要文件和用户登录标识的名称
- 为了利用与用户的经理相关的信息，人员目录应该提供相应的属性，缺省情况下，这是 `manager` 属性
- 为了利用升级电子邮件通知功能，人员目录应该提供用户电子邮件地址

业务流程编排器支持使用下列人员目录来进行人员解析。如果要利用业务流程编排器提供的全套人员分配功能，请使用虚拟成员管理器作为人员目录。

- 联合存储库（也称为虚拟成员管理器）

这是 WebSphere Application Server 支持的缺省人员目录。它提供了对各种目录类型的访问，这些目录类型包括轻量级目录访问协议（LDAP）目录、数据库、基于文件的存储库以及定制存储库。它还支持对存储库进行联合。

既可以检索人员信息，也可以检索组信息。受支持的人员模式（PersonAccount 实体类型）包含用户的名称、登录身份、经理身份和电子邮件地址的相应属性。为了可用于人员解析，必须在 WebSphere Application Server 中将联合存储库配置为活动安全性域定义。

- LDAP 目录

业务流程编排器可以直接访问 LDAP 目录以便进行人员解析，而不必使用 WebSphere Application Server 安全性功能。为了在人员解析（由业务流程编排器实现）和用户认证（由 WebSphere Application Server 实现）之间确保一致性，必须配置 WebSphere Application Server 安全性，以访问业务流程编排器中指定用于人员解析的 LDAP 目录服务器。

根据所使用 LDAP 人员模式的不同，与人员相关的信息包括用户名、身份、经理名和电子邮件地址。为了可用于进行人员解析，业务流程编排器人员目录提供程序配置是必需的。

- WebSphere Application Server 用户注册表

用户注册表是用于检索用户信息的应用程序服务器子系统。业务流程编排器可以将此用户注册表用作人员目录。业务流程编排器使用自己的用户注册表人员目录提供程序来访问 WebSphere Application Server 用户注册表。

相关概念

第 43 页的『升级』

升级是指人员任务在所指定时间内未被处理时自动生成的警报。例如，如果在定义的时间限制内未声明或未完成任务，那么将生成此警报。对于一个任务，可以指定一个或多个升级。这些升级可以并行启动，也可以作为一连串升级启动。

人员目录提供程序与配置

业务流程编排器使用人员目录提供程序作为用于访问人员目录的适配器。要检索用户信息，可以配置虚拟成员管理器、LDAP、用户注册表以及系统人员目录提供程序。

对人员目录提供程序的选择由所需的人员解析支持决定。要利用业务流程编排器提供的全部人员分配功能，请使用虚拟成员管理器。

所有人员目录提供程序都在节点级提供。

虚拟成员管理器人员目录提供程序

虚拟成员管理器人员目录提供程序用于访问 WebSphere Application Server 联合存储库。您可以使用此提供程序来利用下列各方面的人员解析功能：

- 联合存储库功能，其中包括使用各种存储库，例如文件和数据库存储库、LDAP 目录、属性扩展存储库以及存储库的联合
- 升级电子邮件通知
- 缺席者替代
- 所有的预定义人员分配条件

轻量级目录访问协议 (LDAP) 人员目录提供程序

LDAP 人员目录提供程序用于直接访问 LDAP 目录，而不必使用 WebSphere Application Server。在大多数情况下，WebSphere Application Server 安全性域设置为独立 LDAP 注册表，并配置为指向 LDAP 人员目录提供程序所引用的 LDAP 目录。您可以使用此提供程序来利用下列各方面的人员解析功能：

- 升级电子邮件通知
- 所有的预定义人员分配条件

用户注册表人员目录提供程序

可以使用用户注册表人员目录提供程序来通过 WebSphere Application Server 访问下列人员目录：本地操作系统、独立 LDAP 注册表或独立客户注册表。所使用的人员目录取决于应用程序服务器安全性域的配置。您可以使用此提供程序来利用下列各方面的人员解析功能：

- 最大程度地减少了业务流程编排器的人员目录提供程序配置，这是因为存储库由应用程序服务器的安全性域确定
- 一组有限的预定义人员分配条件。用户注册表人员目录提供程序可以解析用户和组，但无法解析职员 - 经理关系、用户属性或电子邮件地址。

系统人员目录提供程序

系统人员目录提供程序提供了有限的人员解析支持。由于系统提供程序只支持硬编码的查询，因此仅适用于测试工作。

所有人员目录配置都要求启用 WebSphere Application Server 管理安全性和应用程序安全性。

每个人员目录提供程序可以与一个或多个人员目录提供程序配置相关联。除 LDAP 人员目录提供程序外的所有配置都立即可供使用。对于虚拟成员管理器人员目录提供程序而言，必须在 WebSphere Application Server 中配置联合存储库功能。对于 LDAP 提供程序配置而言，必须设置必需的连接参数。此外，必须对 LDAP 提供程序配置的变换文件进行定制。

每个配置都由其 Java 命名和目录接口 (JNDI) 名称唯一标识。JNDI 名称是任务模板定义与用于将人员分配解析为任务角色的人员目录配置之间的链接。请使用 WebSphere Integration Developer 来指定任务模板的配置名。如果在运行时使用任务创建 API 来定义任务，那么可以直接在该 API 中指定配置名。不同的任务模板可以引用不同的人员目录配置。

部署任务模板后，人员目录配置名在所部署模板的生存期内保持不变。如果需要更改与模板相关联的人员目录，那么请使用 WebSphere Integration Developer 来更改任务模板定义中定义的人员目录配置的 JNDI 名称，然后再次部署该模板。

相关任务

第 175 页的『配置 LDAP 人员目录提供程序』

通过本任务来配置轻量级目录访问协议 (LDAP) 人员目录提供程序，业务流程编排器将使用此提供程序来执行人员分配，从而确定能够启动流程或者声明活动/任务的人员。

第 173 页的『配置虚拟成员管理器人员目录提供程序』

通过本任务来配置虚拟成员管理器 (VMM) 人员目录提供程序，业务流程编排器

将使用此提供程序来执行人员分配，从而确定能够启动流程或者声明活动/任务的人员。缺省人员目录提供程序已可供使用，仅当您引入了定制人员分配条件时才需要对其进行配置。

将人员分配条件映射到人员查询

部署应用程序时，人员分配条件定义将转换为多组特定于人员目录配置的查询。所生成的人员查询将与任务模板一起存储在业务流程编排器数据库中。

如果使用虚拟成员管理器作为人员目录，那么仅当定义了定制人员分配条件时，才需要更改 XSL 变换文件中的预定义映射。

变换（XSLT）文件包含用于转换人员分配条件的指示信息。每个人员目录配置都与一个变换文件相关联。对于缺省人员目录配置而言，提供了下列变换文件：

- 用于 LDAP 人员目录提供程序的 LDAPTransformation.xml
- 用于虚拟成员管理器人员目录提供程序的 VMMTransformation.xml
- 用于用户注册表人员目录提供程序的 UserRegistryTransformation.xml
- 用于系统人员目录提供程序的 SystemTransformation.xml 和 EverybodyTransformation.xml

在 Windows® 平台上，这些文件在 *install_root*\ProcessChoreographer\Staff 目录中。在 Linux®、UNIX® 和 i5/OS® 平台上，这些文件在 *install_root*/ProcessChoreographer/Staff 目录中。

用于特定人员目录提供程序的人员查询

与人员目录配置相关联的 XSL 变换文件用于生成特定于特定存储库的人员查询。每个查询都可以由相关人员目录提供程序执行，以获取用户标识列表。可供某个人员目录提供程序使用的预定义查询与该提供程序可以执行的调用相对应，因此固定不变。

人员目录提供程序提供的这组特定于存储库的查询与它用于从相应人员目录中检索用户信息的方法相对应。您可以使用这组查询来构造更复杂的查询，如下列示例所示：

- 对查询结果进行组合，以便将各个查询返回的用户标识添加到当前用户标识结果列表。例如，LDAP 人员目录提供程序允许使用下列预定义查询：

- 所指定组的组成员的用户标识列表：

```
<sldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
...
</sldap:usersOfGroup>
```

- 所指定用户的专有名称（DN）：

```
<sldap:user dn="uid=user1,dc=mycomp" .../>
```

- 可以构造一个复杂的查询来获取所指定组的成员用户标识列表以及所指定用户的 DN：

```
<sldap:staffQueries>
  <sldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
    ...
  </sldap:usersOfGroup>
  <sldap:user dn="uid=user1,dc=mycomp" .../>
</sldap:staffQueries>
```

- 从当前结果列表中除去查询结果。例如，以下代码段说明如何从所指定组成员的标识列表中除去“user1”：

```

<slldap:staffQueries>
  <slldap:usersOfGroup groupDN="cn=group1,dc=mycomp" recursive="yes">
    ...
  </slldap:usersOfGroup>
  <slldap:remove value="user1"/>
</slldap:staffQueries>

```

- 使用从一个查询获取的查询结果来影响后续查询的行为。例如，在以下片段中，执行了两个查询。首先，检索用户“uid=user1,...”的 LDAP 条目中的“manager”属性值并将其保存到中间变量“supervisor”。然后，使用此变量来查找经理的 LDAP 条目并检索相关联的用户标识。

```

<slldap:staffQueries>
  <slldap:intermediateResult name="supervisor">
    <slldap:user dn="uid=user1,dc=mycomp" attribute="manager" ... />
  </slldap:intermediateResult>
  <slldap:user dn="%supervisor% .../>
</slldap:staffQueries>

```

人员目录提供程序可以执行根据这些组合规则构造的人员查询。

定制人员分配条件的变换

如果要根据 LDAP 人员和组模式修改人员分配条件的变换，或者要定义定制人员分配条件，那么需要对该变换进行定制。

要使用定制的变换，必须创建变换文件。请勿更改缺省变换文件，并且不要让您的变换文件使用这些缺省变换文件的名称。对于用于 LDAP 或虚拟成员管理器人员目录提供程序的新变换文件，请始终在该文件中包括 **User Records by User ID** 人员分配条件定义。此定义是业务流程编排器所必需的，即使未对人员分配明确指定此条件亦如此。

要使用新的变换文件，请定义指向该文件的新人员目录配置。

相关任务

第 175 页的『配置 LDAP 人员目录提供程序』

通过本任务来配置轻量级目录访问协议 (LDAP) 人员目录提供程序，业务流程编排器将使用此提供程序来执行人员分配，从而确定能够启动流程或者声明活动/任务的人员。

第 173 页的『配置虚拟成员管理器人员目录提供程序』

通过本任务来配置虚拟成员管理器 (VMM) 人员目录提供程序，业务流程编排器将使用此提供程序来执行人员分配，从而确定能够启动流程或者声明活动/任务的人员。缺省人员目录提供程序已可供使用，仅当您引入了定制人员分配条件时才需要对其进行配置。

定义定制人员分配条件:

您可能需要使用自己的条件对预定义人员分配条件集进行扩展。

创建定制人员分配条件时，需要将其包括在下列文件中:

- VerbSet.xml 文件
- 变换文件

要使用定制的变换，必须创建变换文件。请勿更改缺省变换文件，并且不要让您的变换文件使用这些缺省变换文件的名称。

例如，您创建了新条件 Mentor of Employee。此条件与 Manager of Employee 条件相似，但它检索职员顾问的用户标识，而不是检索职员的经理。

1. 在 VerbSet.xml 文件中添加以下 XML 片段：

```
<vs:DefineVerb name='Mentor of Employee'>
  <vs:Description>Assigns the mentor of an employee.
  Supported by sample XSLT files for:
  - LDAP
  </vs:Description>
  <vs:Mandatory>
    <vs:Parameter>
      <vs:Name>EmployeeName</vs:Name>
      <vs:Type>xsd:string</vs:Type>
    </vs:Parameter>
  </vs:Mandatory>
  <vs:Optional>
    <vs:Parameter>
      <vs:Name>Domain</vs:Name>
      <vs:Type>xsd:string</vs:Type>
    </vs:Parameter>
  </vs:Optional>
</vs:DefineVerb>
```

2. 在变换文件中，定义一个新的 XSL 变量，以包含人员顾问的 LDAP 属性：

```
<xsl:variable name="DefaultMentorAttribute">mentor</xsl:variable>
```

3. 在变换文件中添加以下 XML 片段：

```
<!-- Begin template ManagerOfEmployee -->
<xsl:template name="ManagerOfEmployee">
  <sldap:staffQueries>
    <xsl:attribute name="threshold">
      <xsl:value-of select="$Threshold"/>
    </xsl:attribute>

    <sldap:intermediateResult>
      <xsl:attribute name="name">mentorvar</xsl:attribute>
      <sldap:user>
        <xsl:attribute name="dn">
          <xsl:value-of select="staff:parameter[@id='EmployeeName']"/>
        </xsl:attribute>

        <sldap:resultObject>
          <xsl:attribute name="objectclass">
            <xsl:value-of select="$DefaultPersonClass"/>
          </xsl:attribute>
          <xsl:attribute name="usage">simple</xsl:attribute>

          <sldap:resultAttribute>
            <xsl:attribute name="name">
              <xsl:value-of select="$DefaultMentorAttribute"/>
            </xsl:attribute>
            <xsl:attribute name="destination">intermediate</xsl:attribute>
          </sldap:resultAttribute>
        </sldap:resultObject>

        <sldap:user>
          <xsl:attribute name="dn">%mentorvar%</xsl:attribute>
          <xsl:call-template name="ResultObjectSpecForUserData"/>
        </sldap:user>
      </sldap:staffQueries>
</xsl:template>
<!-- End template ManagerOfEmployee -->
```

缺席者替代

替代功能允许您为自己或者所管理的小组成员指定离开设置。替代策略定义如何处理分配给不在场用户的任务和升级。

替代策略在任务模板建模时定义。同一个策略将应用于所有与某个任务模板相关联的任务角色。部署任务模板后，就无法更改该策略。

如果用户不在场，那么将对人员解析结果应用替代策略，以确定代替不在场用户接收工作项的人员。此策略仅适用于具有人员分配条件的任务角色。这意味着，任务发起者、启动者或所有者不能被替代。同样，刷新人员分配条件时，还将刷新替代。

根据特定替代策略的不同，将执行下列操作：

无替代（缺省）

各个用户保持不变

用户不在场时进行替代

- 对于每个在场用户，选择该用户本身。
- 对于每个不在场用户，选择第一个在场的替代者。
- 如果用户及其替代者都不在场，那么应用缺省人员分配规则。

选择在场用户

- 对于每个在场用户，选择该用户本身。
- 不考虑替代者。
- 如果没有任何用户在场，那么选择最初那组用户，即，不考虑他们不在场这一事实。

替代功能要求使用虚拟成员管理器作为人员目录。要使虚拟成员管理器可用于替代功能，必须在 WebSphere Application Server 中将联合存储库配置为活动安全性域。确保在管理控制台中对人员任务管理器启用替代功能。如果将具有非缺省替代策略的任务模板部署到除虚拟成员管理器以外的人员目录提供程序，那么部署将失败。

相关任务

第 180 页的『配置人员替代功能』

通过本任务创建并激活虚拟成员管理器（VMM）属性扩展存储库，以使业务流程编排器支持用户替代功能。

第 320 页的『指定离开设置』

如果您打算离开办公室一段时间，那么请指定替代您完成任务的人员。

第 321 页的『指定用户的离开设置』

如果用户无法处理他们的任务，例如休病假，那么可以指定替代该用户执行任务的人员。

缺省人员分配

如果未定义某些任务角色的人员分配条件，或者人员解析失败或未返回结果，那么将以缺省方式进行人员分配。直接插入任务和独立任务的缺省人员分配并不相同。

直接插入任务

下表列示了直接插入任务的缺省人员分配。

直接插入人员任务及其升级的角色	在任务模型中未定义此角色的情况下 ...	如果人员分配失败...
任务管理员	仅应用继承规则	仅应用继承规则
任务潜在实例创建者	所有用户都成为潜在实例创建者	所有用户都成为潜在实例创建者
任务潜在启动者	所有用户都成为潜在启动者	所有用户都成为潜在启动者
任务潜在所有者	所有用户都成为潜在所有者	管理员成为潜在所有者
任务编辑者	没有编辑者	没有编辑者
任务阅读者	仅应用继承规则	仅应用继承规则
升级接收者	管理员成为升级接收者	管理员成为升级接收者

对于直接插入任务，将应用下列继承规则：

- 流程管理员成为所有直接插入任务、它们的子任务、后续任务和升级的管理员。
- 流程阅读者成为所有直接插入任务、它们的子任务、后续任务和升级的阅读者。
- 任务管理员成为所有子任务、后续任务以及所有这些任务的升级的管理员。
- 任务阅读者成为所有子任务、后续任务以及所有这些任务的升级的阅读者。
- 任何任务角色的成员都成为此任务的升级、子任务和后续任务的阅读者。
- 升级接收者成为已升级任务的阅读者。

独立任务

下表列示了独立任务的缺省人员分配。

独立人员任务及其升级的角色	在任务模型中未定义此角色的情况下 ...	如果人员分配失败...
任务管理员	发起者成为管理员	任务无法启动
任务潜在实例创建者	所有用户都成为潜在实例创建者	所有用户都成为潜在实例创建者
任务潜在启动者	发起者成为潜在启动者	任务无法启动
潜在所有者	所有用户都成为潜在所有者	管理员成为潜在所有者
编辑者	没有编辑者	没有编辑者
阅读者	仅应用继承规则	仅应用继承规则
升级接收者	管理员成为升级接收者	管理员成为升级接收者

对于独立任务，将应用下列继承规则：

- 任务管理员成为所有子任务、后续任务以及所有这些任务的升级的管理员。
- 任务阅读者成为所有子任务、后续任务以及所有这些任务的升级的阅读者。
- 任何任务角色的成员都成为此任务的升级、子任务和后续任务的阅读者。
- 升级接收者成为已升级任务的阅读者。

使用业务流程管理器 API 调用方法时，BPESystemAdministrator 角色的成员具有管理员权限，而 BPESystemMonitor 角色的成员具有读者权限。通过人员任务管理器 API 调用方法时，TaskSystemAdministrator 角色的成员具有管理员权限，而 TaskSystemMonitor 角色的成员具有读者权限。

相关概念

第 56 页的『人员任务的授权角色』

可以对人员任务执行的操作取决于您的授权角色。此角色可以是系统级 J2EE 角色，也可以是基于实例的角色。

管理人员分配条件和人员解析结果

与任务授权角色相关联的人员分配条件在所部署任务模板或任务实例的整个生存期内有效。

如果需要更改人员分配条件，那么必须在 WebSphere Integration Developer 中更改任务定义，然后重新部署该任务模板。

从人员分配条件派生的人员查询作为所部署任务模板或任务实例的组成部分进行存储。在任务执行期间，授权角色要求对相关联的人员查询进行解析。

人员查询结果取决于人员目录的内容，此内容可能会随着时间的推移而变化。例如，在人员组中可能会增加新成员。为了反映人员目录的更改，必须通过下列其中一种方法来刷新人员查询：

- 由管理员明确刷新

管理员可以使用管理控制台或管理命令来刷新人员查询结果。系统提供了用于执行下列操作的命令：

- 同时刷新所有人员查询结果
- 刷新所有与某个任务模板相关联的人员查询结果
- 刷新当前结果中包含特定用户标识的人员查询结果

- 由对已到期人员查询执行的预定刷新操作触发

此方法根据下列参数执行：

- 人员查询结果的超时值 (T_{out})。
- 人员查询的刷新时间表。请使用 WebSphere Application Server 的 CRON 语法来定义时间表，例如，在每个星期一的下午一点或者每个工作日的午夜进行刷新。

下列参数确定如何自动刷新人员查询：

- 第一次运行查询时或者刷新查询时，查询结果将得到一个到期时间戳记 ($t_{exp} = t_{current} + T_{out}$)
- 当查询刷新守护程序被调用时，将再次运行所有其结果已到期的人员查询

可以将超时值设置为大于时间表刷新时间间隔。例如，可以将超时值设置为 24 小时，并将刷新时间间隔设置为 1 小时。这样，可以将人员查询更新操作分散到一天内的各个时段进行，从而避免同时刷新所有人员查询结果造成负担过重。

相关任务

第 261 页的『使用管理控制台刷新人员查询结果』

人员查询的结果是静态的。您可以使用管理控制台来刷新人员查询。

第 279 页的『使用管理命令刷新人员查询结果』

人员查询的结果是静态的。您可以使用管理命令来刷新人员查询。

第 263 页的『使用刷新守护程序刷新人员查询结果』

如果要对所有已到期的人员查询结果自动执行定期刷新，请使用本方法。

共享人员分配

对于特定的任务角色，任务模板的所有实例使用同一个人分配条件。这是因为，所有任务实例都根据同一个任务模板进行实例化。为了避免重新运行人员查询，任务模板的各个任务实例共享查询结果。

仅当人员分配条件定义包含固定参数值时，才能共享结果。这种值（例如组名 `cn=group1`，`cn=groups`）表示相应的人员查询结果相同，而与解析该人员查询时所处的任务实例上下文无关。

如果人员分配条件定义包含替换变量，那么共享范围将缩小为具有相同替换变量值的人员分配。例如，参数值可以依赖于任务输入消息的某些部件。由于不同任务实例的输入消息可以不同，因此人员查询的参数值也会有所不同。

如果对人员查询结果进行后处理，那么缺省情况下无法共享这些结果。为了共享已进行后处理的结果，请在管理控制台中完成下列步骤：

1. 如果业务流程编排器被配置在服务器上，那么请单击**服务器** → **应用程序服务器** → ***server_name***。
2. 如果业务流程编排器被配置在集群中，那么请单击**服务器** → **集群** → ***cluster_name***。
3. 在**业务集成**中，单击**业务流程编排器** → **人员任务管理器** → **[其他属性] 定制属性**。
4. 将 **Staff.PostProcessorPlugin.EnableResultSharing** 定制属性值更改为 `true` 并保存更改。
5. 重新启动服务器或集群，以使更改生效。

相关任务

第 496 页的『创建插件以便对人员查询结果进行后处理』

人员解析返回对特定角色（例如任务的潜在所有者）指定的用户列表。您可以通过创建插件来更改人员解析所返回的人员查询结果。例如，为了改善工作负载平衡，可以通过创建插件从查询结果中除去工作负载较高的用户。

第 2 部分 规划和配置业务流程编排器

第 3 章 规划配置业务流程编排器

规划业务流程编排器的设置和配置参数。

过程

1. 执行『规划拓扑、设置和配置路径』。
2. 根据所选配置路径的不同，执行下列其中一项任务：
 - 对于『基本样本』，执行第 84 页的『规划创建业务流程编排器基本样本配置』。
 - 对于『公司样本』，执行第 85 页的『规划创建包含样本公司的业务流程编排器样本配置』。
 - 对于『非生产部署环境』，执行第 85 页的『规划非生产部署环境配置』。
 - 对于『生产部署环境』，执行第 86 页的『规划使用管理控制台的部署环境向导』。
 - 对于『灵活的定制配置』，执行第 89 页的『规划定制业务流程编排器配置』。

结果

执行第 125 页的第 4 章，『配置业务流程编排器』前所要求规划的内容现已规划完毕。

相关概念

第 111 页的『关于业务流程编排器』

描述业务流程管理器 and 人员任务管理器的特性。

规划拓扑、设置和配置路径

您对拓扑和设置所作的选择将影响可以使用的业务流程编排器配置路径。

关于此任务

不同的配置路径在复杂性、灵活性以及对不同拓扑和数据库的支持方面有所不同。

过程

1. 请注意，您必须在 5 条不同的配置路径之间进行选择。
 - 『基本样本』
 - 『公司样本』
 - 『非生产部署环境』
 - 『生产部署环境』
 - 『灵活的定制配置』

对于大多数配置路径而言，您可以随意选择配置工具。

2. 您应该了解可以用来配置业务流程编排器的不同配置工具。

安装程序或概要文件管理工具

提供了最简易的方法来创建非生产系统，所要求执行的规划工作最少。

- 『基本样本』配置包含下列业务流程编排器组件：

- 业务流程编排器
- 资源管理器
- 观察器和事件收集器
- 『公司样本』配置还包含预先配置了 15 个用户的样本公司的人员目录，并且启用了替代功能和组工作项。
- 『非生产部署环境』配置使您能够方便地在集群中配置业务流程编排器，但是，业务流程编排器不能有自己的数据库，而是使用公共的 WPRCSDB 数据库。

管理控制台的部署环境向导

可以用来创建基于部署环境模式的『生产部署环境』业务流程编排器配置。

管理控制台的“业务流程编排器”配置页面

可以使用此管理控制台页面在服务器或集群中配置『灵活的定制配置』业务流程编排器生产系统。此页面允许设置多个需要进行详细规划的配置参数。此页面无法配置业务流程编排器资源管理器或业务流程编排器观察器；它们有自己的配置页面，或者可以通过运行脚本进行配置。此配置路径最适合于创建生产系统。

bpeconfig.jacl 配置脚本

可以使用此脚本在给定服务器或集群中配置『灵活的定制配置』业务流程编排器生产系统以及所有必需资源。您可以通过交互方式运行此脚本，并且，如果提供了所有必需参数，那么还可以通过批处理方式运行此脚本以自动完成重复的配置工作。它可以创建本地数据库以及必需的消息传递资源，并且可以选择配置业务流程编排器资源管理器和业务流程编排器观察器。对于某些数据库系统，此脚本还可以创建远程数据库。此配置路径最适合于创建生产系统。

3. 请注意，某些配置路径有一些限制，这使它们不适用于生产系统。 例如：
 - 在试验某个样本配置后，必须先将其除去，然后才能创建适用于生产系统的配置。
 - 如果创建使用嵌入式 Derby 数据库或 WPRCSDB 公共数据库的配置，那么此配置不适用于高性能系统。必须先除去此配置，然后才能创建使用独立高性能数据库的新配置。
 - 如果使用 FILESSTORE 或嵌入式 Derby 消息存储器，那么无法将概要文件联合到 Network Deployment 环境。为了能够联合概要文件，必须完全除去业务流程编排器配置，然后创建新配置并使用可以通过远程方式进行访问的数据库作为消息存储器。
4. 标识用于确定所要使用的配置路径的主要条件。 使用下表来确定选项和约束：

表 4. 用于选择配置路径的条件

是否适用于生产系统?	部署目标	业务流程编排器配置	是否可以有独立的 BPEDB 数据库?	消息传递引擎支持的消息存储器	适用的配置、工具和选项
否	独立服务器	基本样本 (未包含样本公司, 并且未启用人员分配和替代功能)	是, 但只支持嵌入式 Derby	只支持嵌入式 Derby	『基本样本』, 使用下列其中一项: • 安装程序 • 概要文件管理工具 选择选项: • 独立服务器概要文件 • 典型 • 启用管理安全性
		包含 15 个用户的样本公司并启用了人员分配和替代功能。 此样本与 WebSphere Integration Developer 中提供的样本完全相同。		嵌入式 Derby、文件存储器或 WPRCSDB	『公司样本』, 使用: • 概要文件管理工具 选择选项: • 独立服务器概要文件 • 高级 • 根据开发模板创建服务器 • 启用管理安全性
	集群	部署环境模式选项: • 远程消息传递和远程支持 • 远程消息传递 • 单个集群	否, 共享 WPRCSDB, 此数据库可以是除嵌入式 Derby 以外的任何数据库	共享 WPRCSDB, 此数据库可以是除文件存储器和嵌入式 Derby 以外的任何受支持数据库	『非生产部署环境』, 使用下列其中一项: • 安装程序 • 概要文件管理工具 选择: 部署环境
是	集群	部署环境模式选项: • 远程消息传递和远程支持 • 远程消息传递 • 单个集群 • 定制	是, 除嵌入式 Derby 以外的任何受支持数据库	除文件存储器和嵌入式 Derby 以外的任何受支持数据库	『生产部署环境』, 使用: • 管理控制台 选择: 部署环境
		灵活的定制配置	是, 任何受支持数据库	除文件存储器和嵌入式 Derby 以外的任何受支持数据库	『灵活的定制配置』, 使用下列其中一项: • bpeconfig.jacl 脚本 • 管理控制台的“业务流程编排器”配置页面
	独立服务器			任何受支持的数据库或文件存储器	

注: 也可以使用任何用于创建生产系统的建议配置路径来创建不适用于生产系统的配置。

请考虑下列选项:

- a. 确定是否正在配置生产系统。通常, 生产系统有高性能、可伸缩性和安全性方面的要求。对于业务流程编排器而言, 生产系统应该有自己的非 Derby 数据库 BPEDB。
- b. 确定业务流程编排器的部署目标将是独立服务器还是集群。

- c. 如果您不想创建生产系统，那么请确定独立服务器上的样本配置能否满足您的需要。如果能够满足需要，那么请确定是否要让此样本包含样本人员目录以启用人员分配和替代功能。

注：在样本人员目录中，联合存储库使用文件注册表，并且所有样本人员的密码都是“wid”。WebSphere 管理用户标识也被添加到该目录中，并使用概要文件创建期间指定的密码。

- d. 如果要在集群中配置业务流程编排器，那么根据性能要求的不同，确定消息传递引擎和支持应用程序（例如业务流程编排器资源管理器、观察器和公共事件基础结构）是有各自的集群还是共享同一个集群。标准部署环境模式为：

远程消息传递和远程支持

使用三个集群。这三个集群分别用于应用程序、消息传递引擎和支持应用程序。

远程消息传递

一个集群用于应用程序和支持功能。第二个集群用于消息传递引擎。

单个集群

只有一个集群，同时用于应用程序、消息传递引擎和支持应用程序。

定制 更灵活的设置。

- e. 确定业务流程编排器是否需要专用的 BPEDB 数据库。
 - f. 确定业务流程编排器消息传递引擎是共享 WPRCSDB 数据库还是有自己的消息存储器，并确定它是使用文件存储器还是使用数据库系统。
 - g. 如果要使用业务流程编排器观察器，那么可以在创建业务流程编排器配置的同时配置观察器，也可以稍后再创建观察器。确定业务流程编排器观察器是使用 BPEDB 数据库还是使用自己的 OBSRVDB 数据库。并且，规划业务流程编排器观察器组件的拓扑。有关更多信息，请参阅第 108 页的『规划业务流程编排器观察器』。
5. 如果要让 WebSphere Portal 访问业务流程编排器，那么请进行规划以便在门户网站服务器上配置业务流程编排器客户机。同样，可以配置业务流程编排器客户机，以使任何定制 WebSphere Process Server 客户机能够访问业务流程编排器。
 6. 如果已启用应用程序安全性，并且长时间运行流程调用远程 EJB 方法，那么配置 CSIv2 入站认证时，必须启用公共安全互操作性 V2 (CSIv2) 身份声明。

结果

您已规划拓扑并确定将要使用的配置路径和配置工具。

规划创建业务流程编排器基本样本配置

用于独立服务器的这个基本样本不包括人员分配。

开始之前

您已执行第 81 页的『规划拓扑、设置和配置路径』，并已选择『基本样本』配置路径。

过程

1. 决定是使用安装程序还是概要文件管理工具来创建样本。在这两种情况下，样本完全相同，唯一的区别是使用的工具不同。
2. 如果您希望人员任务管理器能够发送升级电子邮件，那么请规划下列各项：
 - 如果没有可用的本地简单电子邮件传输协议（SMTP）邮件服务器，那么以后需要将邮件会话更改为指向适当的邮件服务器。
 - 进行规划以便更改电子邮件的发件人地址。否则，它将使用伪发件人地址。
3. 请注意，此样本配置将 WebSphere 管理员用户标识和密码用于各种业务流程编排器用户标识。

规划创建包含样本公司的工作流程编排器样本配置

此样本包含由 15 位人员组成的样本公司，适合于在独立服务器上测试人员分配和替代功能。此样本与 WebSphere Integration Developer 中提供的样本完全相同。

开始之前

您已执行第 81 页的『规划拓扑、设置和配置路径』，并已选择『公司样本』配置路径。

关于此任务

此工作流程编排器样本配置只要求执行最少量的配置工作。

过程

1. 请注意，只能使用概要文件管理工具来创建此样本。要获取此样本，必须选择下列选项：
 - 独立服务器概要文件
 - 高级
 - 根据开发模板创建服务器
 - 启用管理安全性例如，如果未启用管理安全性，那么将无法创建工作流程编排器样本配置。
2. 确定工作流程编排器消息传递引擎是将使用文件存储器、嵌入式 Derby 数据库还是公共数据库 WPRCSDB。
3. 如果您希望人员任务管理器能够发送升级电子邮件，那么请规划下列各项：
 - 如果没有可用的本地简单电子邮件传输协议（SMTP）邮件服务器，那么以后需要将邮件会话更改为指向适当的邮件服务器。
 - 进行规划以便更改电子邮件的发件人地址。否则，它将使用伪发件人地址。
4. 请注意，此样本配置将 WebSphere 管理员用户标识和密码用于各种业务流程编排器用户标识。

规划非生产部署环境配置

进行规划以便使用安装程序或概要文件管理工具来创建基于部署环境模式的工作流程编排器配置。

开始之前

您已执行第 81 页的『规划拓扑、设置和配置路径』，并已选择『非生产部署环境』配置路径。

关于此任务

使用部署环境向导时，必须选择部署环境模式，接着，您有机会更改 WBI_BPC 组件的缺省数据库参数和认证别名以及输入业务流程编排器的其他参数。

过程

1. 确定要使用的部署环境模式:

- 远程消息传递和远程支持
- 远程消息传递
- 单个集群

2. 规划“安全性”步骤中要输入的业务流程编排器 JMS 认证别名的用户名。

3. 规划业务流程编排器上下文根步骤:

业务流程编排器资源管理器上下文根

此设置定义浏览器访问业务流程编排器资源管理器时必须使用的部分 URL。

业务流程编排器观察器上下文根

此设置定义浏览器访问业务流程编排器观察器时必须使用的部分 URL。

4. 规划“业务流程编排器”步骤的安全性参数。业务流程管理器和人员任务管理器都将使用这些用户标识和组:

管理员用户和组

规划用户标识列表和/或组列表，业务管理员角色将映射到这些用户标识和组。

监视员用户和组

规划用户标识列表和/或组列表，业务监视员角色将映射到这些用户标识和组。

JMS API 认证用户和密码

业务流程管理器消息驱动的 Bean 的 RunAs 用户标识。

升级用户认证用户和密码

人员任务管理器消息驱动的 Bean 的 RunAs 用户标识。

5. 如果要使用人员分配功能，那么请执行第 106 页的『规划人员目录提供程序』。

规划使用管理控制台的部署环境向导

对于生产系统而言，请规划业务流程编排器的所有配置参数，包括独立的数据库。对于非生产系统而言，可以使用共享数据库。

开始之前

您已执行第 81 页的『规划拓扑、设置和配置路径』，并已选择『生产部署环境』配置路径。

关于此任务

使用部署环境向导时，必须选择部署环境模式，接着，您有机会更改 WBI_BPC 组件的缺省数据库参数和认证别名以及输入业务流程编排器的其他参数。

过程

1. 如果您没有足够的信息或权限，导致无法独自创建完整的配置，那么请咨询负责配置其他系统部件的人员并与其一起进行规划。 例如：
 - 您可能需要获取有关贵公司 LDAP 服务器的信息，如果该服务器要求进行认证，那么您还需要获取用户标识和授权信息。
 - 如果您无权创建数据库，那么数据库管理员必须参与数据库规划工作，并且需要数据库脚本副本以便进行定制和运行。
2. 执行第 89 页的『规划安全性、用户标识和权限』。
3. 确定要使用的部署环境模式：
 - 远程消息传递和远程支持
 - 远程消息传递
 - 单个集群
 - 定制
4. 如果选择定制部署环境模式：
 - a. 确定是否要安装业务流程编排器资源管理器。 如果要安装此产品，那么请规划它的部署位置。
 - b. 确定是否要安装业务流程编排器事件收集器。 如果要安装此产品，那么请规划它的部署位置。
 - c. 确定是否要安装业务流程编排器观察器。 如果要安装此产品，那么请规划它的部署位置。
 - d. 规划 SCA 绑定的上下文根。
 - e. 规划是否要启用状态观察器和审计日志记录。
5. 如果您计划让下列各项使用专用的数据库：
 - 业务流程编排器的 BPEDB 数据库 - 组件 WBI_BPC。
 - 业务流程编排器观察器的 OBSRVRDB 数据库 - 组件 WBI_BPCEventCollector。
 - 业务流程编排器消息传递引擎的 BPEMEDB 数据库 - 组件 WBI_BPC_ME。

规划每个数据库的下列参数，以后，将在向导的数据库页面上输入这些参数：

数据库实例

数据源的名称，例如 BPEDB、OBSRVRDB 或 BPEMEDB，代替缺省值 WPRCSDB（此值将导致共享公共数据库）。缺省值仅适用于低性能设置。

模式 要使用的数据库模式限定符。

创建表 如果选中此选项，那么第一次访问该数据库时，将自动创建表。要使此选项生效，该数据库必须已存在，并且提供的用于创建该数据源的用户名必须有权在该数据库中创建表和索引。如果未选中此选项，那么将不会自动创建表，您必须通过运行脚本手动创建这些表。对于生产系统而言，请不要选中此选项，并使用提供的 SQL 脚本来设置数据库。

用户名和密码

有权连接到数据库和修改数据的用户标识。如果该用户标识有权在数据库

中创建表和索引，那么可以使用用于自动创建表的选项，并且应用服务包或修订包后将根据需要自动更新数据库模式。

服务器 数据库服务器的地址。请指定主机名或 IP 地址。

提供程序

JDBC 提供程序。

有关规划数据库的更多详细信息，请参阅第 95 页的『规划业务流程编排器的数据库』。

6. 规划“安全性”步骤中要输入的业务流程编排器 JMS 认证别名的用户名。

7. 规划业务流程编排器上下文根步骤:

业务流程编排器资源管理器上下文根

此设置定义浏览器访问业务流程编排器资源管理器时必须使用的部分 URL。

业务流程编排器观察器上下文根

此设置定义浏览器访问业务流程编排器观察器时必须使用的部分 URL。

8. 规划“业务流程编排器”步骤的安全性参数。业务流程管理器和人员任务管理器都将使用这些用户标识和组:

管理员用户和组

规划用户标识列表和/或组列表，业务管理员角色将映射到这些用户标识和组。

监视员用户和组

规划用户标识列表和/或组列表，业务监视员角色将映射到这些用户标识和组。

JMS API 认证用户和密码

业务流程管理器消息驱动的 Bean 的 RunAs 用户标识。

升级用户认证用户和密码

人员任务管理器消息驱动的 Bean 的 RunAs 用户标识。

9. 如果要为人员任务管理器升级配置电子邮件会话，那么请规划“业务流程编排器”步骤的下列参数:

邮件传输主机

简单电子邮件传输协议 (SMTP) 电子邮件服务的主机名或 IP 地址。

邮件传输用户和邮件传输密码

如果邮件服务器不要求进行认证，那么可以不填写这些字段。

业务流程编排器资源管理器 URL

此 URL 用来在生成的电子邮件中提供一个链接，以使接收电子邮件通知的业务管理员可以通过单击该链接在 Web 浏览器中查看相关业务流程或人员任务。

10. 如果要使用人员分配功能，那么请执行第 106 页的『规划人员目录提供程序』。

规划定制业务流程编排器配置

使用管理控制台的“业务流程编排器”配置页面或 `bpeconfig.jacl` 配置脚本来规划用于创建定制配置的配置参数和选项。

开始之前

您已执行第 81 页的『规划拓扑、设置和配置路径』，并已选择『灵活的定制配置』配置路径。

过程

1. 确定您将使用下列哪一项来配置业务流程编排器：
 - 管理控制台的“业务流程编排器”配置页面
 - `bpeconfig.jacl` 配置脚本
2. 如果您没有足够的信息或权限，导致无法独自创建完整的配置，那么请咨询负责配置其他系统部件的人员并与其一起进行规划。 例如：
 - 您可能需要获取有关贵公司 LDAP 服务器的信息，如果该服务器要求进行认证，那么您还需要获取用户标识和授权信息。
 - 如果您无权创建数据库，那么数据库管理员必须参与数据库规划工作，并且需要数据库脚本副本以便进行定制和运行。
3. 『规划安全性、用户标识和权限』
4. 第 95 页的『规划业务流程编排器的数据库』
5. 第 105 页的『规划业务流程管理器和人员任务管理器』
6. 第 106 页的『规划人员目录提供程序』
7. 第 107 页的『规划业务流程编排器资源管理器』
8. 第 108 页的『规划业务流程编排器观察器』
9. 如果您将使用管理控制台的“业务流程编排器”配置页面，那么请确保已规划所有要在配置页面上输入的值，如第 115 页的『业务流程编排器配置』所述。
10. 如果您将使用 `bpeconfig.jacl` 配置脚本：
 - a. 确保已规划所有必须在命令行或批处理文件中指定的选项和参数值。第 140 页的『使用 `bpeconfig.jacl` 脚本配置业务流程编排器』对选项和参数作了概述，第 146 页的『`bpeconfig.jacl` 脚本文件』对选项和参数作了详细描述。
 - b. 如果要使用批处理文件来运行 `bpeconfig.jacl` 配置脚本，那么请创建该批处理文件或 `shell` 脚本。

结果

执行第 125 页的第 4 章，『配置业务流程编排器』前所要求规划的内容现已规划完毕。

规划安全性、用户标识和权限

规划用于配置业务流程编排器的用户标识和权限。

关于此任务

在配置期间，需要使用各种用户标识，并且必须指定在运行时将要使用的其他用户标识。在开始配置业务流程编排器前，务必规划并创建所有用户标识。

对于业务流程编排器样本配置:

您只需有权创建新概要文件。在概要文件管理工具中, 使用选项来创建典型的概要文件, 如果已启用管理安全性, 那么还将配置业务流程编排器样本。不需要进行其他规划, 也不需要其他用户标识, 可以跳过本任务。

对于高安全性配置:

必须详细规划所有用户标识, 如本任务所述。

对于低安全性配置:

如果不需要使用全部安全性功能, 例如非生产系统的情况, 那么可以减少所使用的用户标识数。您必须详细规划所有用户标识, 但可以将某些用户标识用于多种用途。例如, 用来创建数据库模式的数据库用户标识也可以作为在运行时用来连接到数据库的数据源用户名。

如果将使用 `bpeconfig.jacl` 脚本来配置业务流程编排器:

用于运行 `bpeconfig.jacl` 脚本的用户标识必须具有执行该脚本将要执行的配置操作所需的权限, 此外, 也可以指定用户标识作为参数。如果指定用户标识作为参数, 那么必须详细规划所有用户标识。对于可以作为 `bpeconfig.jacl` 脚本参数指定的用户标识, 下表提供了参数名。概要文件必须已存在。如果已启用 WebSphere 管理安全性, 那么配置者角色必须包含可用于调用 `wsadmin` 工具的 WebSphere 管理员用户标识。

过程

1. 打印本页面的硬拷贝, 以便将规划的值填入最后一列。请保留此硬拷贝以便配置业务流程编排器时用作参考, 并请将其保留在记录中以供将来参考。
2. 规划在 WebSphere Process Server 中配置业务流程编排器时使用的用户标识。

表 5. 规划 WebSphere Process Server 的用户标识

用户标识或角色	何时使用此用户标识	此用户标识的用途	此用户标识具有的权限	规划的用户标识
配置业务流程编排器的用户	配置期间	登录到管理控制台和运行管理脚本。	WebSphere 管理员或配置者角色 (如果已启用 WebSphere 管理安全性)。	
		运行 <code>bpeconfig.jacl</code> 脚本以配置业务流程编排器。	运行此脚本时, 还必须提供所选选项所必需的用户标识。有关更多信息, 请参阅第 146 页的『 <code>bpeconfig.jacl</code> 脚本文件』。	

3. 规划哪些人员需要访问 `install_root` 的子目录。如果安全策略不允许这些人员拥有此访问权, 那么需要为他们提供这些目录中文件的副本。

表 6. 规划对 *install_root* 的子目录的访问权

用户标识或角色	何时使用此用户标识	此用户标识的用途	此用户标识具有的权限	规划的用户标识
数据库管理员	配置期间	运行用于设置下列数据库的脚本： BPEDB : 业务流程编排器的数据库。 OBSRVDB : 业务流程编排器观察器的数据库。	如果使用 <code>bpeconfig.jacl</code> 脚本来配置业务流程编排器： 对 <code>bpeconfig.jacl</code> 在以下目录的子目录中生成的 <code>createSchema.sql</code> 脚本（或者此脚本的副本）的读访问权： <ul style="list-style-type: none"> 在 Windows 平台上: <code>profile_root\dbscripts\ProcessChoreographer\</code> 在 Linux、UNIX、i5/OS 平台上以及在 z/OS 平台上的 UNIX 系统服务（USS）中: <code>profile_root/dbscripts/ProcessChoreographer/</code> 	
			如果不使用 <code>bpeconfig.jacl</code> 脚本来配置业务流程编排器： 对以下目录中提供的数据库脚本（或者这些脚本的副本）的读访问权： <ul style="list-style-type: none"> 在 Windows 平台上: <code>install_root\dbscripts\ProcessChoreographer\database_type</code> 在 Linux、UNIX、i5/OS 平台上以及在 z/OS 平台上的 UNIX 系统服务（USS）中: <code>install_root/dbscripts/ProcessChoreographer/database_type</code> 其中, <code>database_type</code> 是下列其中一项： <ul style="list-style-type: none"> DB2 DB2zOSV7 DB2zOSV8 Db2iSeries Derby Informix Oracle SQLServer 	
集成开发者	定制期间	要将人员分配条件与轻量级目录访问协议（LDAP）或虚拟成员管理器（VMM）人员目录提供程序配合使用，必须定制样本 XSL 变换文件的副本。	对 <code>Staff</code> 目录的读访问权，或者对以下目录中文件副本的读访问权： <ul style="list-style-type: none"> 在 Windows 平台上: <code>install_root\ProcessChoreographer\Staff</code> 在 Linux、UNIX 和 i5/OS 平台上: <code>install_root/ProcessChoreographer/Staff</code> 集成开发者还需要对适当的目录具有写访问权，以使定制后的 XSL 变换文件可供服务器使用。	

4. 规划用于创建、配置和访问业务流程编排器所使用的数据库的用户标识。

表 7. 规划 *BPEDB* 数据库的用户标识

用户标识或角色	何时使用此用户标识	此用户标识的用途	此用户标识具有的权限	规划的用户标识
数据库管理员	在配置前	创建 BPEDB 数据库实例。对于 Oracle: 用于创建 BPEDB 数据库。	创建数据库。	

表 7. 规划 BPEDB 数据库的用户标识 (续)

用户标识或角色	何时使用此用户标识	此用户标识的用途	此用户标识具有的权限	规划的用户标识
数据库管理员或者将要运行 bpeconfig.jacl 脚本的管理员	配置期间	除非使用嵌入式 Derby 数据库, 否则您或数据库管理员必须运行业务流程编排器数据库脚本。	对于 BPEDB 数据库: 改变表、进行连接、插入表以及创建索引、模式、表、表空间和视图。	
数据源用户名 如果使用 bpeconfig.jacl 脚本, 那么这是 -dbUser 参数。	配置期间	如果选中 创建表 选项, 那么此用户标识将用于创建数据库表。	要使用 创建表 配置选项, 此用户标识还必须有权对 BPEDB 数据库执行下列操作: 改变表、进行连接、插入表以及创建索引、表和视图。	
	运行时	业务流程管理器和人员任务管理器将使用此用户标识来连接至 BPEDB 数据库。	此用户标识必须有权对 BPEDB 数据库执行下列操作: 进行连接、删除表、插入表、选择表和视图以及更新表。	
	在应用服务包或修订包后	必要时, 应用服务后将自动更新数据库模式。仅当此用户标识具有必需的数据库权限时才会进行自动更新, 否则, 必须以手动方式更新模式。	此用户标识必须有权对 BPEDB 数据库执行下列操作: 改变表、创建表、插入表、选择表、连接至该数据库以及创建和删除索引和视图。	

5. 如果您将配置业务流程编排器观察器, 那么请规划用来创建、配置和访问数据库的用户标识。

表 8. 规划 OBSRVDB 数据库的用户标识

用户标识或角色	何时使用此用户标识	此用户标识的用途	此用户标识具有的权限	规划的用户标识
数据库管理员	在配置前	创建 OBSRVDB 数据库实例。对于 Oracle: 用于创建 OBSRVDB 数据库。	创建数据库。	
数据库管理员或管理员	配置期间	运行 setupEventCollector 工具或者用于创建模式的 SQL 脚本。	对于 OBSRVDB 数据库: 改变表、进行连接、创建函数、插入表以及创建索引、模式、表、表空间和视图。 如果将要使用用户定义函数的 Java 实现, 那么该用户标识还必须有权安装 JAR 文件。	
事件收集器的数据源用户名	运行时	连接到观察器数据库。如果正在使用业务流程编排器观察器并且它使用 BPEDB 数据库, 那么请使用业务流程编排器数据源的用户名。	连接至该数据库	

6. 如果要想让业务流程编排器的消息传递引擎消息存储库使用独立的数据库, 而不是使用嵌入式 Derby 数据库或文件存储器, 那么请规划用于访问该数据库的用户标识。

表 9. 规划预先配置的 BPEME 消息传递引擎数据库的用户标识

用户标识或角色	何时使用此用户标识	此用户标识的用途	此用户标识具有的权限	规划的用户标识
总线数据源用户名 如果使用 bpeconfig.jacl 脚本，那么这是 -medbUser 参数。	配置期间以及运行时	此用户名用于连接至 BPEME 消息传递总线的数据库以及创建必需的表和索引。	此用户标识必须有权对 BPEME 数据库执行下列操作：进行连接、删除表、插入表、选择表和视图以及更新表。	

7. 规划 Java 消息服务 (JMS) 提供程序的用户标识。

表 10. 规划 JMS 提供程序的用户标识

用户标识或角色	何时使用此用户标识	此用户标识的用途	此用户标识具有的权限	规划的用户标识
JMS 认证用户	运行时	系统集成总线的认证别名。在配置业务流程编排器时，必须指定此别名。 如果使用 bpeconfig.jacl 脚本，那么此用户标识及其密码是参数 -mqUser 和 -mqPwd。	此用户名必须有效。它将被自动添加至业务流程编排器总线的“总线连接者”角色。	
JMS API 认证用户	运行时	将使用此用户标识来处理所有业务流程管理器 JMS API 请求。 如果使用 bpeconfig.jacl 脚本，那么此用户标识及其密码是参数 -jmsBFMRUnAsUser 和 -jmsBFMRUnAsPwd。	这些标识必须有效。	
升级认证用户	运行时	将使用此用户标识来处理所有人员任务管理器 JMS API 请求。 如果使用 bpeconfig.jacl 脚本，那么此用户标识及其密码是参数 -jmsHTMRUnAsUser 和 -jmsHTMRUnAsPwd。		

8. 规划业务流程管理器和人员任务管理器的 J2EE 角色所映射的组或用户标识。

表 11. 规划业务流程管理器和人员任务管理器的安全角色

用户标识或角色	何时使用此用户标识	此用户标识的用途	规划的用户标识和/或组列表
管理员用户	运行时	业务流程管理器和人员任务管理器的系统管理员和监视员安全角色都映射到用户标识和/或组列表。此处定义的值用于创建映射，以使具有此角色的用户拥有他们所需的访问权。 如果使用 bpeconfig.jacl 脚本，那么这些用户和组与下列参数相对应： • -adminBFMUsers 和 -adminHTMUsers • -adminBFMGroups 和 -adminHTMGroups • -monitorBFMUsers 和 -monitorHTMUsers • -monitorBFMGroups 和 -monitorHTMGroups	
管理员组	运行时		
监视员用户	运行时		
监视员组	运行时		

9. 如果您希望人员任务升级发送特定业务事件的电子邮件通知，并且简单电子邮件传输协议（SMTP）服务器要求进行认证，那么请确定用来连接至电子邮件服务器的用户标识。

表 12. 规划电子邮件服务器的用户标识

用户标识或角色	何时使用此用户标识	此用户标识的用途	此用户标识具有的权限	规划的用户标识
邮件传输用户	运行时	人员任务管理器使用此用户标识向已配置的邮件服务器进行认证，以便发送升级电子邮件。 如果使用 bpeconfig.jacl 脚本，那么这是 -mailUser 参数。密码是 -mailPwd 参数。	发送电子邮件。	

10. 如果将使用人员任务的人员分配功能，并且要使用的轻量级目录访问协议（LDAP）人员目录提供程序采用简单认证方法，那么请规划用于登录到 LDAP 服务器的用户标识。

表 13. 规划 LDAP 服务器的用户标识

用户标识或角色	何时使用此用户标识	此用户标识的用途	此用户标识具有的权限	规划的用户标识
LDAP 插件属性: 认证别名	运行时	在配置使用简单认证方法来连接至 LDAP 的轻量级目录访问协议（LDAP）人员目录提供程序时，例如，mycomputer/My LDAP Alias。在定制 LDAP 插件的属性时，请指定此用户标识。	如果 LDAP 服务器使用简单认证方法，那么此用户标识必须能够连接至 LDAP 服务器。如果 LDAP 服务器使用匿名认证方法，那么不需要此用户标识。	

11. 创建已规划所需权限的用户标识。如果您自己无权创建所有这些用户标识，那么请向相应的管理员提交请求，并在此表中填写要创建的用户标识的名称。

结果

您已确定配置业务流程编排器时需要使用的用户标识。

规划业务流程编排器的数据库

规划业务流程编排器的数据库。根据设置的不同，可能需要进行规划以便创建 0 到 3 个数据库。

关于此任务

业务流程编排器可以与其他流程服务器组件共享数据库。BPEDB 数据库由业务流程管理器 and 人员任务管理器使用。对于生产系统，请进行规划，以便让每个配置了业务流程编排器的部署目标使用专用的数据库。如果您正在使用业务流程编排器观察器，那么它可以使用同一个 BPEDB 数据库，但使用独立的 OBSRVDB 数据库可以提高性能。业务流程编排器消息传递引擎可以共享 SCA 消息传递引擎所使用的数据库，也可以使用它们自己的 BPEMEDB 数据库。有关所选配置路径所支持的数据库的更多信息，请参阅第 83 页的表 4。

过程

1. 对于简化设置比性能更为重要的非生产系统而言，选项取决于所选配置路径：
 - 如果您将使用安装程序或概要文件管理工具来创建『基本样本』业务流程编排器配置，那么请创建独立的 Derby 数据库 BPEDB，此数据库也由业务流程编排器观察器使用。对于业务流程编排器消息传递引擎，缺省情况是使用独立的 Derby 数据库 BPEME，但如果您使用『公司样本』配置，那么还可以选择使用文件存储器或共享的 WPRCSDB 数据库。
 - 如果您将使用安装程序或概要文件管理工具来创建包含业务流程编排器配置的部署环境，那么业务流程编排器、业务流程编排器观察器和业务流程编排器消息传递引擎都将使用 WPRCSDB 存储器。因此，不需要为业务流程编排器执行任何数据库规划工作。
2. 对于生产系统：
 - a. 如果性能至关重要，那么请进行规划，以便让业务流程编排器使用独立的数据库，如『规划 BPEDB 数据库』所述。否则，请规划使用 WPRCSDB 公共数据库。
 - b. 如果您将使用业务流程编排器观察器：
 - 如果要最大程度地降低查询对业务流程性能产生的影响，请规划使用独立的数据库，如第 100 页的『规划 OBSRVDB 数据库』所述。
 - 否则，请进行规划，将其配置为使用 BPEDB 数据库。
 - c. 如果要想让业务流程编排器消息传递引擎使用独立的数据库，以便提高性能，那么请执行第 104 页的『规划消息传递引擎数据库』，否则请规划使用由服务组件体系结构（SCA）使用的缺省数据库。

结果

业务流程编排器配置的所有数据库已规划完毕。

规划 BPEDB 数据库

规划业务流程编排器的数据库。

关于此任务

业务流程编排器需要使用数据库。系统为所有受支持的数据库系统提供了用于创建和管理数据库模式的 SQL 脚本。数据库一旦就绪，就必须对业务流程编排器进行配置，

以便通过 JDBC 访问该数据库。根据数据库系统、拓扑、安装目的以及您选择使用的管理工具的不同，可以自动执行某些或全部用于创建数据库和配置 JDBC 访问的任务。对于生产系统而言，业务流程编排器应该有自己的数据库。但是，如果性能并不重要，那么还可以将业务流程编排器配置为与其他 WebSphere Process Server 组件共享数据库。

过程

1. 确保选择的 BPEDB 数据库与配置路径兼容。支持下列数据库：

- DB2 UDB Linux 版、UNIX 版和 Windows 版
- DB2 iSeries™ 版
- DB2 z/OS® 版
- Derby
- Informix Dynamic Server
- Microsoft® SQL Server
- Oracle

如果已确定业务流程编排器的配置方式，那么选择的配置路径将对数据库的创建方式产生影响。如果尚未确定用来配置业务流程编排器的配置路径，那么确定数据库要求将有助于避免选择不支持您的需求的配置路径。有关每条配置路径所支持的数据库的详细信息，请参阅第 83 页的表 4。

2. 如果您希望使用能够最大程度减少规划工作并具有下列特征的简单数据库设置：

- 不具备生产系统通常所需的性能、可伸缩性以及安全性。
- 所有数据库对象都在同一个表空间（例如缺省表空间）中创建。
- 数据库服务器在 WebSphere Process Server 机器本地。
- 用于访问数据库的用户还具有管理权限。

所需规划的选项随所选配置路径的不同而有所变化：

- 如果使用**安装程序**或**概要文件管理工具**来获取业务流程编排器样本配置，那么将为业务流程编排器创建独立的 Derby 数据库 BPEDB，这不要求进行进一步的规划工作。
- 如果使用管理控制台的“**部署环境**”向导来配置业务流程编排器，那么请进行规划，以使用所提供的 SQL 脚本的副本来创建 BPEDB 数据库，这将在单一的表空间中创建缺省模式。
- **bpeconfig.jacl** 工具可以配置业务流程编排器以及对任何数据库的 JDBC 访问。
 - 如果将以交互方式运行 bpeconfig.jacl 脚本，那么可以选择在现有数据库中创建表。
 - 如果您的用户标识有权创建数据库对象，那么可以使用 -createDB yes 选项。此选项将使 bpeconfig.jacl 脚本生成并运行用于在缺省表空间中创建数据库对象的 SQL 文件。在这种情况下，还需要规划停止服务器并对 wsadmin 实用程序指定 -conntype NONE 选项。如果您正在使用 Oracle 或 DB2 z/OS 版数据库，那么数据库实例必须已存在。对于其他数据库类型而言，bpeconfig.jacl 将尝试创建数据库实例。如果创建数据库或对象期间发生任何错误，那么可以使用所生成的 SQL 脚本，就像指定了 -createDB no 选项一样。
 - 如果您的用户标识无权创建数据库对象，那么必须使用 -createDB no 选项。此选项将使 bpeconfig.jacl 脚本生成用于在缺省表空间中创建数据库对象的 SQL 文件，但不会运行该脚本。在这种情况下，应该请数据库管理员为您定制和运行该脚本。

有关此工具和其他数据库参数的更多信息，请参阅第 146 页的『bpeconfig.jacl 脚本文件』。

• 使用管理控制台的“**业务流程编排器**”配置页面:

- 要在公共数据库中创建业务流程编排器的数据库对象，请使用缺省数据库实例作为业务流程编排器数据源的目标。
- 要重复使用现有的数据库，请指定现有数据库实例作为业务流程编排器数据源的目标。
- 如果选中创建表选项，那么业务流程编排器第一次使用该数据库时，将在该数据库中创建所需的数据库对象。此选项不适用于 DB2 z/OS 版数据库。
- 要使用脚本来创建数据库，请不要使用创建表选项。

3. 如果您希望使用具有下列特征的**高性能**业务流程编排器数据库设置:

- 数据库仅由业务流程编排器使用。
- 理想情况下，数据库服务器在专用机器上，但也可以在 WebSphere Process Server 机器本地。
- 您可以定制表空间在磁盘上的分配方式，以提高性能。
- 可以使用两个不同的用户标识来分别访问和管理数据库。

您必须规划下列所有步骤:

4. 如果尚未规划数据库的用户标识，那么请执行第 91 页的表 7。

5. 规划磁盘和表空间的分配方式。在理想情况下，数据库主机应该具有快速存储子系统，例如与网络连接的存储器或存储区域网络。对于生产系统而言，应该考虑开发和系统测试期间取得的经验。数据库的大小取决于多种因素。作为微流程运行的进程使用非常少量的空间，但每个流程模板可能需要数十或数百个千字节的空间。

如果将使用独立的磁盘，并且数据库系统支持将数据库表分配到不同的磁盘，那么请规划要使用的磁盘数以及分配方式。硬件辅助磁盘阵列的性能通常优于单个的磁盘。如果您使用下列其中一项:

- DB2
- DB2zOSV7
- DB2zOSV8
- Informix (已知表空间名为 dbspaces)
- Oracle

规划下列 BPEDB 数据库表空间的存储位置:

- AUDITLOG
- COMP
- INDEXTS
- INSTANCE
- LOBTS
- STAFFQRY
- TEMPLATE
- WORKITEM
- SCHEDTS

它们可以在一个高性能 RAID 阵列中，但是，每个表空间应该在各自的文件中，以便进行并行访问。记住，对于给定数目的磁盘，使用 RAID 配置时的性能优于将表空间分配给独立磁盘时的性能。例如，对于在带有 N 个处理器的专用服务器上运行的 DB2 数据库，请考虑遵循下列准则：

- 对于表空间，请使用带有 2*N 个主磁盘、2*N 个镜像磁盘并且条带大小为 256 kb 的 RAID-1 阵列。
- 对于数据库事务日志，请使用带有 2*N 个主磁盘、2*N 个镜像磁盘并且条带大小为 64 kb 的 RAID-1 阵列。

如果正在使用 DB2 数据库，并且将在 RAID 控制器上使用 15 个磁盘驱动器，那么请考虑采用以下分配方式：

- 一个磁盘用于操作系统和页面调度（在 Windows 上称为页面文件，在 AIX[®] 和 HP-UX 上称为调页空间，在 Solaris 上称为交换空间）。
- 采用 RAID-1 配置，使用 8 个磁盘（4 个主磁盘，4 个镜像磁盘）作为一个逻辑磁盘来存储数据库控制文件和表空间，并使用条带大小 256 kB。
- 采用 RAID-1 配置，使用 6 个磁盘（3 个主磁盘，3 个镜像磁盘）作为一个逻辑磁盘来存储数据库事务日志，并使用条带大小 64 kB。

如果正在使用 Oracle 数据库，那么请考虑下列准则：

- 使用一兆字节条带宽度，对所有磁盘上的所有文件进行“完全分块并镜像”（SAME）。
 - 对数据进行镜像，以实现高可用性。
 - 在外面那一半的磁盘驱动器上创建供表空间使用的分区。
 - 按分区而不是按磁盘对数据划分子集。
 - 使用自动存储器管理（ASM）文件系统。
 - 不要将重做日志与其他数据文件分开。
6. 请进行规划，在运行用于创建数据库对象的 SQL 脚本前，由您或数据库管理员对其进行定制。
- 如果使用 **bpeconfig.jacl** 工具来配置业务流程编排器，那么请使用 **-createDB no** 选项。这将使此工具不运行它所生成的 SQL 脚本。所生成的 SQL 文件基于系统为数据库提供的标准 SQL 文件，但在该 SQL 文件中预先填充了所有您对 **bpeconfig.jacl** 提供的配置参数，这将最大程度地减少所需进行的定制工作。有关此工具和其他数据库参数的更多信息，请参阅第 140 页的『使用 **bpeconfig.jacl** 脚本配置业务流程编排器』。
 - 如果使用管理控制台的“**业务流程编排器**”配置页面或“**部署环境**”向导来配置业务流程编排器，那么请规划清除创建表选项，以确保不会创建缺省模式。并且，请进行规划，以便对产品为数据库系统提供的标准 SQL 文件的副本进行定制。

如果要预览数据库的 SQL 文件，以便可以规划所要进行的定制工作，请找到并查看数据库的 SQL `createSchema.sql` 脚本，但不要进行修改。这些 SQL 文件的位置是：

- 在 Windows 平台上：`install_root\dbscripts\ProcessChoreographer\database_type`
- 在 Linux、UNIX、i5/OS 平台上以及在 z/OS 平台上的 UNIX 系统服务（USS）中：`install_root/dbscripts/ProcessChoreographer/database_type`

其中，`database_type` 是下列其中一项：

- DB2
 - DB2zOSV7
 - DB2zOSV8
 - Db2iSeries
 - Derby
 - Informix
 - Oracle
 - SQLServer
7. 如果数据库服务器不在流程服务器本地，那么请规划在流程服务器上安装 Java 数据库连接（JDBC）驱动程序或数据库客户机：
 - 对于 2 类 JDBC 驱动程序：确定要安装的数据库客户机以及安装位置。
 - 对于 4 类 JDBC 驱动程序：找到驱动程序的 JAR 文件并确定安装位置。
 8. 如果数据库服务器在流程服务器本地，那么访问数据库时所需的 JDBC JAR 文件已经与数据库系统安装在一起。请查找并记录这些 JAR 文件的位置。
 9. 如果使用 DB2 z/OS 版，那么请确定要使用的子系统。规划脚本文件 createTablespace.sql 和 createSchema.sql 中存储器组名、数据库名（而不是子系统名）和模式限定符的替换值。
 10. 确定用于主管数据库的服务器。对于远程数据库服务器而言，需要相应的数据库客户机或支持 XA 的 4 类 JDBC 驱动程序。
 11. 确定需要对数据库指定的下列配置参数的值：
 - Java 数据库连接（JDBC）提供程序可以是 2 类或 4 类提供程序。对于 Oracle 而言，请确定是使用 oci 还是 thin 驱动程序。
 - 数据库实例（对于 Oracle，这是数据库名称；对于 DB2 z/OS 版，这是子系统名称）。
 - 模式限定符。缺省情况是，使用连接用户标识作为隐式模式限定符。
 - 用于创建该模式的用户名。
 - 如果正在使用 4 类 JDBC 驱动程序：数据库服务器的名称或 IP 地址。
 - 数据库服务器使用的端口号。仅当正在使用 4 类 JDBC 驱动程序时，此端口号才是必需的。
 - 认证别名的用户标识和密码。这是 jdbc/BPEDB 数据源在运行时用于访问该数据库的用户标识。用户标识和密码分别是 bpeconfig.jacl 的 -dbUser 和 -dbPwd 参数。

有关这些参数的更多信息，包括缺省值，请参阅第 115 页的『业务流程编排器配置』或第 146 页的『bpeconfig.jacl 脚本文件』中的描述。
 12. 进行规划，以便支持足够的并行 JDBC 连接：
 - a. 估算业务流程编排器 BPEDB 数据库所需的并行 JDBC 连接的最大数目。此数目取决于业务流程的性质以及用户数。正确的估算如下所示：可以通过业务流程编排器 API 同时连接的最大客户机数加上 JMS 激活规范 BPEInternalActivationSpec 和 HTMLInternalActivationSpec 中定义的并发端点数，再加上 10% 的安全边际以应对超负荷状态。
 - b. 确保数据库能够支持所需的并行 JDBC 连接数。
 - c. 根据数据库系统的最佳做法来规划合适的设置，以支持期望的并行 JDBC 连接数。
 13. 对于生产系统，请规划下列管理任务：

- 在数据库包含典型的生产数据后，对其进行调整。
- 从数据库中定期删除已完成的流程实例和任务实例。

结果

业务流程编排器数据库已规划完毕。

相关任务

第 544 页的『平衡硬件资源』

通过平衡硬件资源，可以提高长时间运行业务流程的性能。

规划 OBSRVRDB 数据库

为业务流程编排器观察器规划数据库。

关于此任务

业务流程编排器观察器可以与其他组件使用同一个数据库，但使用独立的数据库 OBSRVRDB 可以提高性能。如果您不想重复使用 BPEDB 数据库，那么请执行下列操作：

过程

1. 如果您计划使用多个事件收集器实例，并且它们将使用同一个数据库，那么请为每个事件收集器规划唯一的模式名。为了提高性能，请为每个事件收集器规划一个数据库。
2. 确定数据库的数据库系统：
 - Derby
 - DB2 UDB Linux 版、UNIX 版和 Windows 版
 - DB2 iSeries 版
 - DB2 z/OS 版
 - Oracle

限制： 业务流程编排器观察器不支持使用 Informix 或 SQL Server 数据库。

3. 确定用于主管数据库的服务器。
4. 如果尚未规划数据库的用户标识，那么请执行第 92 页的表 8。
5. 如果您不准备使用 bpeconfig.jacl 脚本将观察器和事件收集器配置为使用 BPEDB 数据库，那么请确定如何创建 OBSRVRDB 数据库。

使用菜单驱动的管理工具 `setupEventCollector`

您可以使用此工具以交互方式创建数据库，这将根据运行时环境验证您的输入。如果使用此工具，那么请确定是否要让此工具创建但不运行 SQL 文件 - 如果要在运行此 SQL 文件前对其进行定制，或者要将其提供给数据库管理员以进行定制并运行，那么请使用此选项。有关此工具的更多信息，请参阅第 238 页的『`setupEventCollector` 工具』。

此工具允许您创建基于 Java 的用户定义函数 (UDF) 或基于 SQL 的 UDF，并允许在这两个选项之间进行切换以及安装和除去支持 UDF 所需的 JAR 文件。对于非 Derby 数据库，此工具支持使用基于 Java 的用户定义函数 (UDF) 或基于 SQL 的 UDF 来创建数据库。对于 DB2 z/OS 版数据库，此工具支持使用基于 Java 的 UDF 或基于 SQL 的 UDF 来创建数据库。对于 Derby 数据库而言，仅使用基于 Java 的 UDF 来创建数据库。

运行 SQL 脚本

如果您无权使用工具来访问数据库，那么需要使用 SQL 脚本。对于非 Derby 数据库而言，所有 SQL 脚本都将使用 SQL 实现为业务流程编排器观察器创建 UDF，这简化了使用基于 SQL 的 UDF 进行配置时的任务。对于 Derby 数据库而言，仅使用基于 Java 的 UDF 来创建数据库。

在第一次使用时自动创建表

如果您希望轻松获得缺省数据库模式，请在管理控制台的“业务流程编排器”配置页面上选中**创建表**选项。此选项不适用于高性能系统。对于非 Derby 数据库而言，将使用基于 SQL 的 UDF，这简化了配置任务。此选项不适用于 DB2 z/OS 版数据库。对于 Derby 数据库而言，仅使用基于 Java 的 UDF 来创建数据库。

注：如果使用 DB2 z/OS 版数据库，并且希望使用基于 Java 的 UDF 来创建该数据库，而不是使用基于 SQL 的 UDF，那么只能使用菜单驱动的管理工具 setupEventCollector。因为嵌入式 Derby 数据库不支持 SQL UDF，因此使用 Derby 数据库时，将使用基于 Java 的 UDF。

有关 UDF 的更多信息，请参阅第 222 页的『业务流程编排器观察器的用户定义函数』。

6. 如果使用 DB2 Linux 版、UNIX 版或 Windows 版数据库，那么请规划下列各项：
 - 数据库名称。缺省值是 BPEDB。
 - 用于连接至该数据库的用户标识。您还必须知道此用户标识的密码。
 - 要在其中创建数据库对象的数据库模式的名称。缺省值是连接的用户标识。
 - 规划表空间 OBSVRTS 的标准位置。
 - 确定是否要使用基于 SQL 的用户定义函数（UDF）以代替缺省的基于 Java 的 UDF。
 - 如果将使用 setupEventCollector 工具来设置数据库，那么还需规划下列各项：
 - 确定要使用的 JDBC 驱动程序的类型：
 - 2 类，使用本机数据库客户机进行连接。这是缺省设置。
 - 4 类，直接通过 JDBC 进行连接。在这种情况下，您还需确保了解下列各项：
 - 数据库服务器的主机名或 IP 地址。缺省值是 localhost。
 - 数据库使用的端口号。缺省值是 50000。
 - 找到 DB2 JDBC 驱动程序文件、db2jcc.jar 和 db2jcc_license_cu.jar 的安装目录。
7. 如果使用 DB2 i5/OS 版数据库，那么请规划下列各项：
 - 数据库名称。如果在本机 iSeries 环境（例如 Qshell）中配置数据库，那么请使用 *LOCAL。否则，请使用 *SYSBAS。
 - 用于连接至该数据库的用户标识。您还必须知道此用户标识的密码。
 - 要在其中创建数据库对象的数据库模式的名称。缺省值是连接的用户标识。
 - 确定是否要使用基于 SQL 的用户定义函数（UDF）以代替缺省的基于 Java 的 UDF。
 - 如果将使用 setupEventCollector 工具来设置数据库，那么还需规划下列各项：
 - 数据库服务器的主机名。通常，此值始终是 localhost。端口号始终是 446。

- JDBC 驱动程序的目录:
 - 如果数据库在本机 i/Series 环境中, 例如在 Qshell 中, 那么此目录是 db2_classes.jar 文件所在的路径, 该路径通常是 /QIBM/ProdData/Java400/ext。
 - 对于远程数据库而言, 这是 jt400.jar 文件所在的路径。
8. 如果使用 DB2 z/OS 版数据库, 那么请规划下列各项:
- 子系统的位置名 (网络名)。
 - 存储器组名。
 - 子系统识别的数据库名称。缺省值是 OBSVRDB。
 - 用于连接至该数据库的用户标识。您还必须知道此用户标识的密码。
 - 要在其中创建数据库对象的数据库模式的名称 (SQLID)。
 - 根据所使用 DB2 版本的不同, 请规划将在哪个存储器组中创建表空间:
 - 对于 DB2 z/OS 版版本 7:
 - 对于 OBSVRTS, 使用常规表空间。
 - 对于 OS26201、OS26202、OS26203 和 OS26204, 使用大对象 (LOB) 表空间。
 - 对于 DB2 z/OS 版版本 8:
 - 对于 OBSVR01、OBSVR02、OBSVR03、OBSVR04、OBSVR05、OBSVR06、OBSVR07 和 OBSVR08, 使用常规表空间。
 - 对于 OS26201、OS26202、OS26203 和 OS26204, 使用 LOB 表空间。
 - 如果要使用基于 Java 的用户定义函数 (UDF) 来代替缺省的 SQL 用户定义函数, 那么请确定用于运行函数的 WLM 环境的名称。
 - 如果将使用 setupEventCollector 工具来设置数据库, 那么还需规划下列各项:
 - 确定要使用的 JDBC 驱动程序的类型:
 - 4 类, 直接通过 JDBC 进行连接。在这种情况下, 您还需确保了解下列各项:
 - 数据库服务器的主机名或 IP 地址。缺省值是 localhost。
 - 数据库使用的端口号。缺省值是 446。
 - JDBC 驱动程序 JAR 文件 db2jcc.jar 和 db2jcc_license_cisuz.jar 的目录。
 - 2 类, 使用本机数据库客户机进行连接。在这种情况下, 还需规划本地目录中的数据库别名。
9. 如果使用 Derby 数据库, 那么请规划下列各项:
- 数据库名称。此名称必须是服务器文件系统中的标准路径。缺省值是 *install_root/databases/BPEDB*。
 - 要在其中创建数据库对象的数据库模式的名称。缺省值是 APP。
 - 如果将使用 setupEventCollector 工具来设置数据库, 那么还需规划下列各项:
 - 如果使用 Derby Network JDBC 驱动程序, 那么请规划用于连接数据库的用户标识。您还必须知道此用户标识的密码。
 - 确定要使用的 JDBC 驱动程序的类型:
 - 嵌入式 JDBC 驱动程序。在这种情况下, 还需规划 JDBC 驱动程序 JAR 文件 derby.jar 的目录。缺省位置是 *install_root/derby/lib*。

- 网络 JDBC 驱动程序。在这种情况下，您还需确保了解下列各项：
 - JDBC 驱动程序 JAR 文件 `derbyclient.jar` 的目录。缺省位置是 `install_root/derby/lib`。
 - 如果正在使用 Derby 网络服务器，那么请确定 Derby 网络服务器上 UDF JAR 文件 `bpcodbutil.jar` 的位置。缺省位置是 `install_root/derby/lib`。
 - 数据库服务器的主机名。缺省值是 `localhost`。
 - 数据库使用的端口号。缺省值是 `1527`。
10. 如果使用 Oracle 数据库，那么请规划下列各项：
- SID 名称。缺省值是 `BPEDB`。
 - 确定用于连接至该数据库的 Oracle 用户标识。此用户标识必须具有 `CONNECT` 和 `RESOURCE` 角色。缺省用户标识为 `system`。您还必须知道此用户标识的密码。
 - 要在其中创建数据库对象的数据库模式的名称。缺省值是用于连接至该数据库的用户标识。
 - 规划下列每个表空间的标准位置：
 - `OBSVRIDX`
 - `OBSVRLOB`
 - `OBSVRTS`
 - 确定是否要使用基于 SQL 的用户定义函数（UDF）以代替缺省的基于 Java 的 UDF。
 - 如果将使用 `setupEventCollector` 工具来设置数据库，那么还需规划下列各项：
 - JDBC 驱动程序文件 `ojdbc14.jar` 的位置。
 - 数据库服务器的主机名。缺省值是 `localhost`。
 - 数据库使用的端口号。缺省值是 `1521`。
11. 如果以批处理方式使用 `bpeconfig.jacl` 工具并选中 `-createEventCollector yes` 选项，那么请规划下列其中一项：
- `-createDB yes` 选项使 `bpeconfig.jacl` 运行此工具生成的 SQL 脚本。您可以使用 `-dbSchema` 参数来指定 `BPEDB` 数据库的模式限定符，但业务流程编排器观察器将使用同一个数据库中的同一个模式。这就是 `-createDB yes` 选项不适用于高性能系统的原因。
 - `-createDB no` 选项使此工具不运行它所生成的 SQL 脚本。所生成的 SQL 文件基于系统为数据库提供的标准 SQL 文件，但在该 SQL 文件中预先填充了所有您对 `bpeconfig.jacl` 提供的配置参数，这将最大程度地减少所需进行的定制工作。请进行规划，在运行用于创建数据库对象的所生成 SQL 脚本前，由您或数据库管理员对其进行定制。有关此工具和其他数据库参数的更多信息，请参阅第 140 页的『使用 `bpeconfig.jacl` 脚本配置业务流程编排器』。
- 限制：** 无法以交互方式使用 `bpeconfig.jacl` 脚本来配置业务流程编排器观察器。
12. 如果您将使用管理控制台的“**业务流程编排器事件收集器**”页面来创建数据库表，那么请规划下列其中一项：
- 对于除 `DB2 z/OS` 版以外的所有数据库类型，可以使用创建表选项，以使此工具在业务流程编排器第一次访问所指定数据库时在该数据库中创建缺省模式。

- 如果要运行 SQL 脚本以准备数据库表，那么请不要使用创建表选项。请进行规划，在运行用于创建数据库对象的 SQL 脚本副本前，由您或数据库管理员对其进行定制。此选项最适合于生产系统。
13. 如果要预览数据库的 SQL 脚本，以便可以规划所需进行的定制工作，请执行下列操作：找到并查看数据库的 `createSchema_Observer.sql` 文件，但不要进行修改。这些 SQL 文件的位置是：
- 在 Windows 平台上：`install_root\dbscripts\ProcessChoreographer\database_type`
 - 在 Linux、UNIX、i5/OS 平台上以及在 z/OS 平台上的 UNIX 系统服务（USS）中：`install_root/dbscripts/ProcessChoreographer/database_type`

其中，`database_type` 是下列其中一项：

- DB2
- DB2zOSV7
- DB2zOSV8
- Db2iSeries
- Derby
- Oracle

注：如果使用 `bpeconfig.jacl` 工具来配置业务流程编排器，那么请规划使用此工具生成的 SQL 脚本，您不需要编辑此脚本以将配置参数占位符替换为值。生成的脚本仅在运行此工具后才可用，但它们基于上述位置中的脚本。如果要对表空间的分配进行定制，那么必须对生成的脚本文件进行编辑。

结果

业务流程编排器观察器的数据库已规划完毕。

规划消息传递引擎数据库

通过让业务流程编排器总线的消息传递引擎使用独立的数据库，可以提高性能。

关于此任务

可以让 WebSphere Process Server 创建的所有总线（即服务组件体系结构（SCA）系统总线、SCA 应用程序总线、公共事件基础结构总线和业务流程编排器总线）使用同一个消息传递数据库。为了确保能够对消息传递引擎进行故障转移，该数据库应该可供该消息传递引擎所在集群内的所有成员访问。如果性能至关重要，那么请进行规划，以便让业务流程编排器消息传递引擎使用专用数据库，而不是使用 SCA 总线和应用程序所使用的缺省数据库 MEDB。

过程

1. 如果使用**安装程序**或**概要文件管理工具**来获取其中一个业务流程编排器样本配置，那么请确定业务流程编排器消息传递引擎是将使用嵌入式 Derby 数据库、文件存储器还是 WPRCSDB 数据库。
2. Java 数据库连接（JDBC）提供程序。请注意，文件存储器和嵌入式 Derby 数据库不适用于 Network Deployment 环境。
3. 如果要使用 WebSphere MQ，那么必须使用 `bpeconfig.jacl` 配置脚本来配置业务流程编排器。不推荐使用 WebSphere MQ。
4. 如果使用 `bpeconfig.jacl` 配置脚本在独立服务器上配置业务流程编排器，那么请确定要让消息存储器使用下列哪一项：

- SCA 的消息传递引擎数据库中的独立模式。
 - 独立的 Derby 数据库。
 - 运行 bpeconfig.jacl 脚本前在部署目标上创建的用户定义数据源。
 - 文件存储器。
5. 如果使用管理控制台的“业务流程编排器”配置页面，那么请规划下列配置参数。有关这些参数的更多信息，请参阅“业务流程编排器”配置页面中有关第 121 页的『总线』的部分。
 - 本地或远程总线成员位置。
 - 数据库的名称。缺省值为 BPEME。
 - 模式名。缺省值为 MEDBPM00。
 6. 如果正在使用文件存储器或嵌入式 Derby JDBC 提供程序，那么将自动创建消息存储器。
 7. 如果未使用文件存储器或嵌入式 Derby JDBC 提供程序，那么请规划下列配置参数。
 - a. 进行规划，确保该数据库在业务流程编排器启动前已存在。
 - b. 数据库服务器的主机名或 IP 地址以及它使用的端口号。
 - c. 用于连接到数据库以及用于创建模式的用户名。这是您在第 93 页的表 9 中规划的用户标识。

结果

业务流程编排器消息传递引擎的数据库已规划完毕。

规划业务流程管理器和人员任务管理器

业务流程编排器配置的核心由业务流程管理器和人员任务管理器组成。您必须规划它们的配置参数。

过程

1. 确保您知道要用作业务流程管理器消息驱动的 Bean 的 RunAs 用户标识的 Java 消息服务 (JMS) 提供程序用户标识。在管理控制台以及第 93 页的表 10 中，此标识称为 **JMS API 认证用户**。
2. 确保您知道要用作人员任务管理器消息驱动的 Bean 的 RunAs 用户标识的 Java 消息服务 (JMS) 提供程序用户标识。在管理控制台以及第 93 页的表 10 中，此标识称为 **升级用户认证用户**。
3. 确保您知道管理员和监视员的安全角色所映射的组或用户标识。有关详细信息，请参阅第 94 页的表 11。
4. 如果要让人员任务管理器发送升级事件的电子邮件通知，那么请指定简单电子邮件传输协议 (SMTP) 电子邮件服务器所在主机的主机名或 IP 地址。请规划用于发送电子邮件通知的发件人地址。如果电子邮件服务要求进行认证，那么请确保您知道用于连接到该服务的用户标识和密码。
5. 确定 API 的 Web Service 绑定的上下文根。
 - 在服务器上进行配置时:
 - 业务流程管理器的缺省值是 /BFMIF_nodeName_serverName。
 - 人员任务管理器的缺省值是 /HTMIF_nodeName_serverName。
 - 在集群中进行配置时:

- 业务流程管理器的缺省值是 `/BFMIF_clusterName`。
 - 人员任务管理器的缺省值是 `/HTMIF_clusterName`。
6. 确定最初是否要对业务流程管理器和/或人员任务管理器启用审计日志记录。
 7. 如果您将要使用业务流程编排器观察器，那么确定最初是否要将业务流程管理器配置为生成公共事件基础结构日志记录事件。

结果

业务流程管理器和人员任务管理器的所有初始配置参数现已规划完毕。以后，您随时可以使用管理控制台更改这些设置。

规划人员目录提供程序

规划业务流程编排器的人员目录提供程序、人员替代、虚拟成员管理器和轻量级目录访问协议（LDAP）设置。

过程

1. 如果将要使用人员任务，那么请确定要使用的人员目录提供程序：

虚拟成员管理器（VMM）人员目录提供程序

VMM 人员目录提供程序可以使用您为 WebSphere 安全性预先配置的联合存储库（也称为虚拟成员管理器）- 使用文件存储器。如果要将另一个用户存储库与联合存储库配合使用，那么需要重新配置联合存储库。VMM 人员目录提供程序支持包括替代功能在内的所有业务流程编排器人员分配功能。它依赖于联合存储库提供的功能，例如对各种存储库类型（例如 LDAP、数据库、基于文件和属性扩展存储库）的支持。

要使用 VMM 人员目录提供程序，必须配置用于 WebSphere Application Server 安全性功能的联合存储库。可以使联合存储库与其他基于 LDAP 或数据库的存储库相关联。有关这方面的更多信息，请参阅在联合存储库配置中管理域。有关使用联合存储库的更多信息，请访问 IBM WebSphere Developer 技术日志。

轻量级目录访问协议（LDAP）人员目录提供程序

要使用此人员目录提供程序，必须先对其进行配置。请执行步骤 2 中的规划工作。

系统人员目录提供程序

在使用此人员目录提供程序前，无需对其进行配置。此提供程序仅适用于应用程序开发测试工作，请勿将其用于生产系统。

用户注册表人员目录提供程序

在使用此人员目录提供程序前，无需对其进行配置。

2. 如果将要使用轻量级目录访问协议（LDAP），请规划下列各项。
 - a. 您可能需要定制自己的 `LDAPTransformation.xml` 文件版本。要了解该文件的位置以及可能需要定制的属性列表，请参阅第 175 页的『配置 LDAP 人员目录提供程序』。
 - b. 规划下列 LDAP 定制属性：

LDAP 插件属性	必需或可选	描述
AuthenticationAlias	可选	用于连接到 LDAP 的认证别名，例如 mycomputer/My LDAP Alias。必须在管理控制台中定义此别名：单击安全性 → 安全管理、应用程序和基础结构 → Java 认证和授权服务 → J2C 认证数据。如果未设置此别名，那么将以匿名方式登录到 LDAP 服务器。
AuthenticationType	可选	如果此属性设置为 simple（表示进行简单认证），那么必须指定 AuthenticationAlias 参数。否则，如果未设置此属性，那么将进行匿名认证。
BaseDN	必需	用于所有 LDAP 搜索操作的专有名称（DN），例如 "o=mycompany, c=us"。要指定目录根，请指定空字符串 ""。
Casesentiveness ForObjectclasses	可选	确定 LDAP 对象类的名称是否区分大小写。
ContextFactory	必需	设置 Java 命名和目录接口（JNDI）上下文工厂，例如 com.sun.jndi.ldap.LdapCtxFactory。
ProviderURL	必需	此 Web 地址必须指向 LDAP JNDI 目录服务器和端口。格式必须采用正常 JNDI 语法，例如 ldap://localhost:389。对于 SSL 连接，请使用 LDAP 的 URL。
SearchScope	必需	所有搜索操作的缺省搜索范围。确定在 baseDN 属性下方的搜索深度。请指定下列其中一个值：objectScope、onelevelScope 或 subtreeScope。
additionalParameter Name1-5 和 additionalParameter Value1-5	可选	使用这些“名称/值”对来设置多达 5 个用于连接到 LDAP 服务器的任意 JNDI 属性。

3. 如果将要使用虚拟成员管理器，请规划下列各项。
 - a. 您可能需要定制自己的 VMMTransformation.xml 文件版本。要了解该文件的位置以及可能需要定制的属性列表，请参阅第 173 页的『配置虚拟成员管理器人员目录提供程序』。
4. 如果要在生产环境中使用人员替代功能，那么请规划使用 VMM 属性扩展存储库来存储替代信息。这要求使用一个数据库。属性扩展存储库以及隐式选择的数据库在整个 WebSphere 单元中必须唯一并且可访问。由于 BPEDB 数据库在单元中不一定唯一，因此不能使用 BPEDB。可以使用公共数据库 WPSRCDB 来存放属性扩展存储库，但是，对于生产环境，建议您使用独立于其他 WebSphere Process Server 数据库的数据库。

结果

人员目录提供程序和人员分配选项已规划完毕。

规划业务流程编排器资源管理器

规划业务流程编排器资源管理器的配置参数。

关于此任务

如果您将使用业务流程编排器资源管理器，那么可以在配置业务流程编排器的同时对其进行规划和配置，也可以稍后再执行配置工作。

过程

1. 确定要配置的业务流程编排器资源管理器实例数。配置业务流程编排器时，您可以方便地创建第一个实例。使用多个实例的可能原因包括：

- 由于每个业务流程编排器资源管理器实例只能管理一个业务流程编排器配置，所以，如果环境中有多业务流程编排器配置，那么必须为每个配置设置一个业务流程编排器资源管理器实例。
- 如果您将要定制业务流程编排器资源管理器，那么您可能希望使用两个或更多个不同定制版本的业务流程编排器资源管理器来管理同一个业务流程编排器配置。

如果需要其他实例，那么必须使用管理控制台（如第 123 页的『业务流程编排器资源管理器设置』所述）或配置脚本（如第 189 页的『使用 `clientconfig.jacl` 脚本文件配置业务流程编排器资源管理器』所述）逐个进行创建。

2. 对于所需的每个业务流程编排器资源管理器实例，请规划下列各项：

- 业务流程编排器资源管理器的上下文根。此上下文根在单元中必须唯一。缺省值为 `/bpc`。
- 要插入到升级电子邮件中的业务流程编排器资源管理器 URL。

如果使用业务流程编排器资源管理器来管理同一个业务流程编排器配置，那么业务流程编排器配置的人员任务管理器只能链接至其中一个业务流程编排器资源管理器实例。在这种情况下，必须确定人员任务管理器将要使用的业务流程编排器资源管理器 URL。

- 查询返回的最大结果数 - 缺省值是 10000。
- 此业务流程编排器资源管理器将要管理的业务流程编排器实例的部署目标（服务器或集群）。

有关这些配置参数的更多信息，请参阅第 123 页的『业务流程编排器资源管理器设置』。

结果

业务流程编排器资源管理器的配置选项已规划完毕。

规划业务流程编排器观察器

进行规划，以便配置业务流程编排器观察器和事件收集器。

关于此任务

如果将要使用业务流程编排器观察器，那么可以在配置业务流程编排器的同时对其进行规划和配置，也可以稍后再执行配置工作。

过程

1. 了解不同业务流程编排器观察器拓扑元素的用途以及它们之间的关系。

观察器应用程序

可以在一个服务器或集群中配置多个实例。这些实例不必在已配置业务流程编排器的部署目标上。每个实例都通过一个数据源连接到刚好一个数据库模式。

事件收集器应用程序

此应用程序必须部署在配置了公共事件基础结构（CEI）服务器的服务器或

集群中。它不必部署在配置了业务流程编排器的目标上。它接收来自 CEI 的业务流程事件，对这些事件进行转换，然后将它们写入 OBSRVDB 数据库。

OBSRVDB 数据库

事件收集器和观察器通过同一个数据库进行通信。对于非生产系统而言，可以与其他组件共享数据库。

您所选择的业务流程编排器观察器与业务流程编排器设置的拓扑无关。有关这两者可能相关的更多洞察信息，请参阅第 112 页的『关于业务流程编排器观察器』。

2. 确定设置用途、机器要求和拓扑含义。

简单设置

为了简化配置和管理工作，请将观察器和事件收集器应用程序部署在配置了业务流程编排器和 CEI 的部署目标上，并使用本地数据库系统，但此配置的性能欠佳。

高负载生产系统: Network Deployment

使用包含多个节点的单元，并使用多个集群。将观察器应用程序的实例安装在单元中的任何部署目标上。将事件收集器应用程序安装在配置了公共事件基础结构 (CEI) 的集群中。使用独立的数据库服务器。

3. 如果尚未规划观察器的数据库，那么请执行第 100 页的『规划 OBSRVDB 数据库』。
4. 对于要配置的每个事件收集器实例，请规划下列各项:
 - a. 确定安装位置。在每个部署目标上只能安装一个事件收集器实例，并且必须在部署目标上配置 CEI。
 - b. 确定将如何配置此事件收集器实例:
 - 使用管理控制台页面。有关此选项的更多信息，请参阅第 226 页的『使用管理控制台配置业务流程编排器事件收集器』。
 - 使用交互式 `setupEventCollector` 工具。有关此选项的更多信息，请参阅第 224 页的『使用 `setupEventCollector` 工具配置业务流程编排器事件收集器』。
 - 在创建业务流程编排器配置的同时，使用 `bpeconfig.jacl` 脚本。`-createEventCollector` 选项的缺省值为 `yes`。

注: 请不要使用 `bpeconfig.jacl` 为高性能系统配置业务流程编排器观察器，这是因为 `bpeconfig.jacl` 将在业务流程编排器配置所在的部署目标上配置事件收集器和观察器应用程序，并且将它们配置为共享 BPEDB 数据库。有关此选项的更多信息，请参阅第 140 页的『使用 `bpeconfig.jacl` 脚本配置业务流程编排器』。

无法以交互方式使用 `bpeconfig.jacl` 来配置业务流程编排器观察器。

- c. 规划数据源:
 - 如果安全性至关重要，那么请进行下列规划:
 - 如果业务流程编排器观察器与业务流程编排器共享同一个物理数据库，那么请将观察器数据库规划为使用独立的数据源，并规划它的 JNDI 名称。
 - 规划数据库要使用的认证别名。
 - 如果安全性并不重要，那么请进行下列规划:
 - 如果业务流程编排器观察器与业务流程编排器共享同一个物理数据库 (BPEDB)，那么可以使用同一个数据源，并因此使用同一个 JNDI 名称。

- 如果使用同一个数据源，那么还将使用同一个认证别名。
 - 进行规划以便创建具有单元作用域的数据源。
- d. 规划配置事件收集器时所需的配置参数:
- 数据库的 JNDI 数据源名称。
 - 数据库对象使用的模式。缺省值是用于连接至该数据库的用户标识。
 - 用于连接至该数据库的用户标识。缺省值是 db2admin。
 - 该用户标识的密码。
 - 如果正在使用 4 类 JDBC 连接，那么还需收集数据库服务器的主机名/IP 地址以及它使用的端口号。
 - 确定事件收集器的部署位置。必须在部署目标上配置 CEI，因此，如果 CEI 在独立的集群中，那么请将事件收集器规划为部署在同一个集群中。
 - 如果将在 Network Deployment 环境中部署事件收集器，那么您必须知道哪个部署目标上配置了 CEI 总线的消息传递引擎。
 - 如果对 CEI 总线启用了安全性，那么请规划用于向 CEI 总线进行认证的 JMS 用户标识。
 - 确定是在配置事件收集器时启用 CEI 事件日志记录业务事件，还是稍后使用管理控制台或通过运行脚本来启用这些事件。
- e. 规划运行时配置值，在配置事件收集器后，可能需要对这些配置值进行定制以使其符合您的需要:
- BpcEventTransformerEventCount
 - BpcEventTransformerMaxWaitTime
 - BpcEventTransformerToleranceTime
 - ObserverCreateTables
 - 如果认证别名用户标识将不会拥有该数据库模式，那么请规划 ObserverSchemaName。

有关这些值的更多信息，请参阅第 234 页的『更改业务流程编排器观察器的配置参数』。

5. 对于所配置的每个业务流程编排器观察器实例，请规划下列各项:
- 确定将如何配置此实例:
 - 使用管理控制台页面。有关此选项的更多信息，请参阅第 231 页的『使用管理控制台配置业务流程编排器观察器』。
 - 使用交互式 setupObserver 工具。有关此选项的更多信息，请参阅第 230 页的『使用 setupObserver 工具配置业务流程编排器观察器』。
 - 在创建业务流程编排器配置的同时，使用 bpeconfig.jacl 脚本。-createObserver 选项的缺省值为 yes。

注: 请不要使用 bpeconfig.jacl 为高性能系统配置业务流程编排器观察器，这是因为 bpeconfig.jacl 将在业务流程编排器配置所在的部署目标上配置事件收集器和观察器应用程序，并且将它们配置为共享 BPEDB 数据库。有关此选项的更多信息，请参阅第 140 页的『使用 bpeconfig.jacl 脚本配置业务流程编排器』。

- 此实例的部署目标。
- 此实例将监视的事件收集器所使用数据源的 JNDI 名称。

- 如果将使用管理控制台来配置此实例，那么还需规划下列各项：
 - 此实例的上下文根。此内容指定浏览器访问此业务流程编排器观察器实例时使用的部分 URL。缺省值是 /bpcobserver。
 - 事件收集器实例的部署目标，此业务流程编排器观察器实例将对该事件收集器实例的数据进行可视化。

有关这些配置参数的更多信息，请参阅第 124 页的『业务流程编排器观察器设置』。

6. 如果将使用 bpeconfig.jacl 脚本来配置业务流程编排器：
 - 缺省情况下，以批处理方式运行此脚本时，它还将配置事件收集器和观察器应用程序，并且将在业务流程编排器配置所在的部署目标上配置这些应用程序。
 - 如果您不希望 bpeconfig.jacl 配置事件收集器和/或观察器应用程序，那么请规划使用 bpeconfig.jacl 选项 -createEventCollector no 和/或 -createObserver no，以使 bpeconfig.jacl 不配置这些应用程序。

结果

业务流程编排器观察器和事件收集器的配置选项已规划完毕。

关于业务流程编排器

描述业务流程管理器 and 人员任务管理器的特性。

业务流程编排器是企业工作流程引擎，它支持 WebSphere Application Server 环境中的业务流程和人员任务。这些构造可以用来协调服务以及将涉及人员的活动集成到业务流程中。业务流程编排器可以管理业务流程和人员任务的生命周期、对相关模型进行导航并调用适当的服务。

业务流程编排器具有下列特性：

- 支持业务流程和人员任务。业务流程提供了使用 Web Service 业务流程执行语言（WS-BPEL，缩写为 BPEL）设计业务流程的标准方法。借助人员任务，可以使用任务执行语言（TEL）对涉及人员的活动进行建模。业务流程和人员任务都作为面向服务体系结构（SOA）或服务组件体系结构（SCA）中的服务发布；它们还支持简单数据对象和业务对象。
- 提供了应用程序编程接口，用于开发与业务流程和人员任务进行交互的定制应用程序。
- 提供了业务流程编排器资源管理器。这个 Web 应用程序提供了用于管理业务流程和人员任务的功能。有关更多信息，请参阅『关于业务流程编排器资源管理器』。
- 提供了业务流程编排器观察器。这个 Web 应用程序使您能够观察运行中的业务流程的状态。有关更多信息，请参阅第 112 页的『关于业务流程编排器观察器』。

相关任务

规划配置业务流程编排器

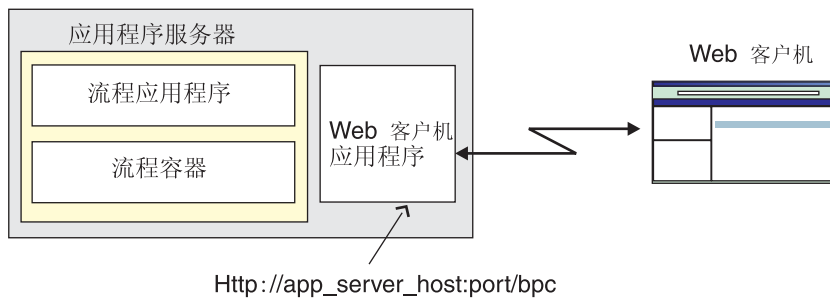
规划业务流程编排器的设置和配置参数。

关于业务流程编排器资源管理器

业务流程编排器资源管理器是一个 Web 应用程序，它实现了用于与业务流程和人员任务进行交互的通用 Web 用户界面。

可以在服务器或集群中配置一个或多个业务流程编排器资源管理器实例。使用带有 WebSphere Process Server 概要文件的 WebSphere Process Server 安装或者 WebSphere Process Server 客户机安装已足够 - 不必在服务器或集群中配置业务流程编排器。客户机安装是将客户机连接至 WebSphere Process Server 时所需的唯一基础结构，但它未包含业务流程编排器资源管理器。但是，如果有 Deployment Manager，那么还可以将业务流程编排器资源管理器安装在 WebSphere Process Server 客户机安装中的服务器上。

一个业务流程编排器资源管理器只能连接至一个业务流程编排器配置，但不必连接至本地配置。但是，可以在同一服务器或集群中配置多个业务流程编排器资源管理器实例，并且每个实例都可以连接至不同的业务流程编排器配置。



启动业务流程编排器资源管理器之后，用户界面中显示的对象以及可以执行的操作取决于您所属的用户组以及该组被授予的权限。例如，如果您是业务流程管理员，那么对所部署的业务流程的平稳运行负责。您可以查看有关流程模板、任务模板、流程实例、任务实例及其相关信息。您还可以对这些对象执行操作；例如，可以启动新的流程实例、创建和启动任务、修复和重新启动失败的活动、管理工作项以及删除已完成的流程实例和任务实例。但是，如果您是用户，那么只能查看那些分配给您的任务并对它们执行操作。

关于业务流程编排器观察器

关于业务流程编排器观察器。

可以使用业务流程编排器观察器来创建有关已完成的流程的报告。并且，还可以使用它来查看运行中流程的状态。本节描述体系结构以及可能的配置路径。

业务流程编排器观察器使用公共事件基础结构（CEI）来收集 WebSphere Process Server 发出的事件。您可以使用众多预定义的报告，也可以定义自己的报告，以获取有关流程数、活动数或其他聚集数据的概览。您还可以获取有关特定流程或活动的信息。

业务流程编排器观察器基于两个 J2EE 企业应用程序，如下图所示：

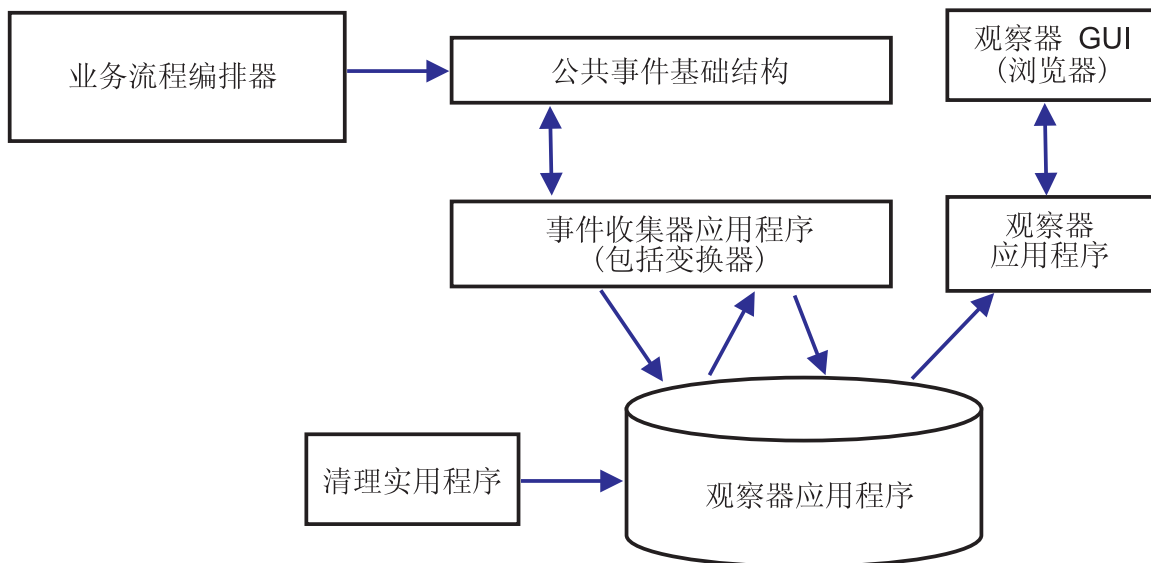


图 1. 业务流程编排器观察器体系结构

- 事件收集器应用程序从 CEI 总线读取事件信息并将其存储在业务流程编排器观察器数据库的事件收集器表中。
- 观察器数据库是一组用于存储事件数据的数据库表。
- 事件变换器 J2EE 企业应用程序定期触发，它将原始事件数据转换为适合于从业务流程编排器观察器中进行查询的格式。
- 观察器应用程序生成报告并执行其他操作，用户可以使用观察器的图形用户界面（GUI）来启动这些操作。
- 您可以使用 GUI 来生成报告。此外，还可以存储和检索已定义的报告。
- 可以使用清理实用程序从观察器数据库中除去记录，这有助于提高性能。

简单配置

下图说明了一种简单配置，在此配置中，性能不是重要的考虑事项。

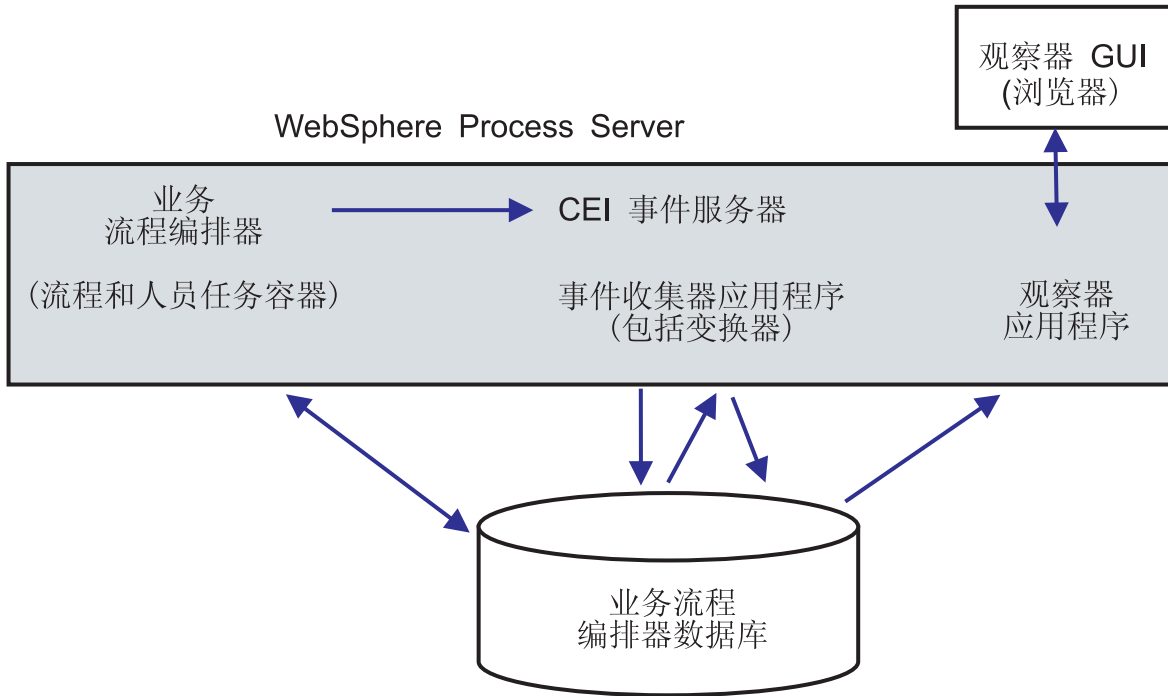


图 2. 业务流程编排器观察器独立设置

所有内容安装在一台机器上，业务流程编排器与业务流程编排器观察器使用同一个数据库。

在创建业务流程编排器样本配置时，将创建这种简单配置。并且，在缺省情况下，`bpeconfig.jacl` 工具将在业务流程编排器配置所在的部署目标上配置观察器和事件收集器应用程序。将启用公共事件基础结构 (CEI) 日志记录，并且将在业务流程编排器 Derby 数据库 `BPEDB` 中创建必需的数据库模式。如果性能并不重要，那么此配置路径比较理想。

高性能配置

系统提供了交互式配置工具，这些工具使您能够随意地利用业务流程编排器观察器体系结构的全部潜能。例如，在生产系统的理想性能配置中，三个业务流程编排器元素在不同的机器上运行，并且业务流程编排器观察器使用自己的数据库。

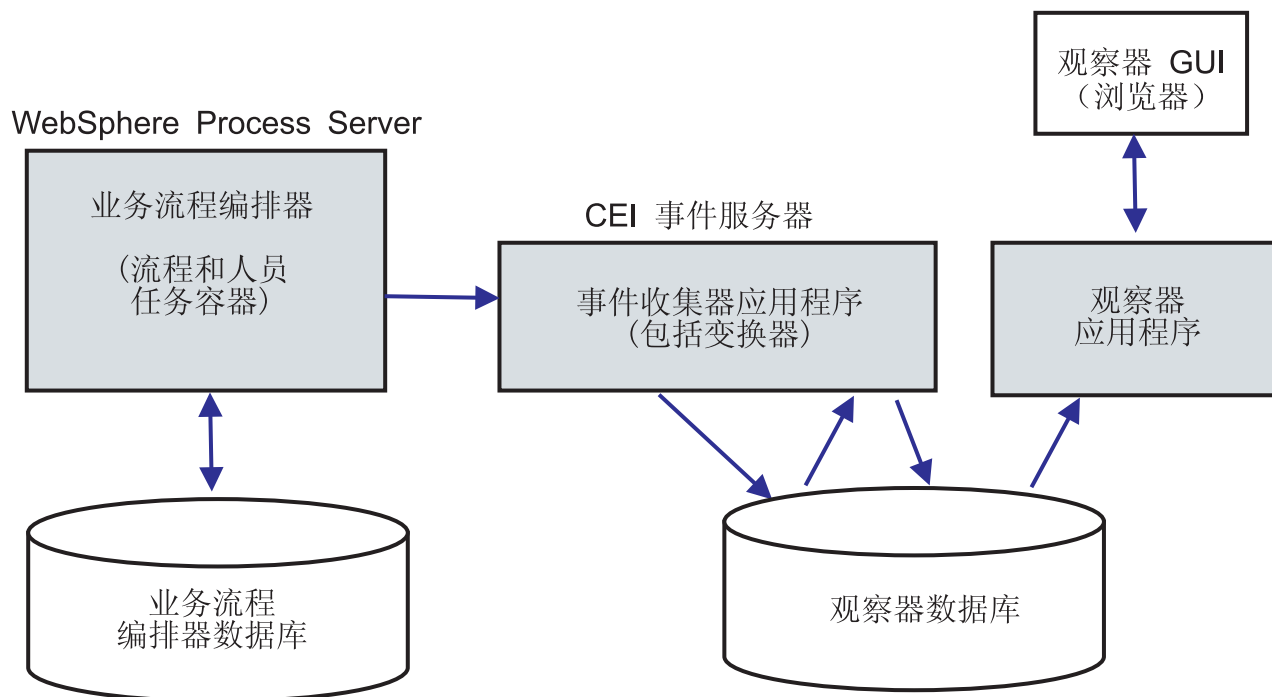


图 3. 能够提高生产性能的业务流程编排器观察器设置

如果要想业务流程编排器观察器使用独立的数据库，或者要在集群式设置中将业务流程编排器观察器添加到现有的业务流程编排器配置或者使用更复杂的数据库选项，那么请执行第 191 页的『配置业务流程编排器观察器』。

在 Network Deployment 环境中

如果要在 Network Deployment 环境中配置业务流程编排器观察器，那么存在下列约束。

- 必须在单元中配置 CEI。
- 如上图所示，必须在 CEI 事件服务器配置所在的部署目标上配置业务流程编排器事件收集器。如果所配置的 CEI 事件服务器与业务流程编排器不在同一个集群中，那么必须在 CEI 事件服务器配置所在的部署目标上配置业务流程编排器事件收集器。业务流程编排器观察器应用程序不需要与事件收集器安装在同一台机器上。

业务流程编排器配置

使用此面板来安装和配置业务流程编排器。

要查看此管理控制台页面，请单击服务器 → 集群 → *cluster_name* 或服务器 → 应用程序服务器 → *server_name*，然后在容器中单击业务流程编排器容器。

此页面分为多个部分。有关每一部分中各个字段的详细信息，请参阅：

- 第 116 页的『数据源』
- 第 117 页的『人员任务管理器邮件会话』
- 第 118 页的『安全性』
- 第 120 页的『状态观察器』
- 第 120 页的『SCA 绑定』

- 第 121 页的『总线』

数据源

在此部分中，指定业务流程编排器的数据源。

编辑

单击此项以编辑数据源。

测试连接

测试与数据源的连接。

数据库实例

业务流程管理器和人员任务管理器使用的数据库的名称。

属性	值
数据类型	字符串
缺省值	WPRCSDB

模式名

要使用的模式的名称。

仅当您想使用自己的模式来代替缺省模式时，才需要更改模式名。

属性	值
数据类型	字符串

创建表

如果选中此选项，那么第一次访问该数据库时，将自动创建表。要使此选项生效，该数据库必须已存在，并且提供的用户名必须有权在该数据库中创建表和索引。

对于生产系统而言，建议您不要使用此选项。如果未选中此选项，那么将不会自动创建表，您必须通过运行脚本手动创建这些表。

属性	值
数据类型	复选框
缺省值	选中

用户名

有权连接到数据库和修改数据的用户标识。

如果该用户标识有权在数据库中创建表和索引，那么可以使用用于自动创建表的选项，并且应用服务包或修订包后将根据需要自动更新数据库模式。

属性	值
数据类型	字符串

密码

数据源用户标识的密码。

属性
数据类型

值
字符串

服务器

数据库服务器的地址。

指定主机名/IP 地址以及端口号。

属性
数据类型
示例

值
字符串
localhost:50000

提供程序

业务流程编排器的 JDBC 提供程序。

属性
数据类型

值
下拉列表

人员任务管理器邮件会话

在此部分中，指定升级电子邮件的参数。

启用电子邮件服务

如果人员任务管理器将发送升级电子邮件通知，那么必须启用邮件会话。

属性
数据类型
缺省值

值
复选框
选中

邮件传输主机

简单电子邮件传输协议（SMTP）电子邮件服务的主机名或 IP 地址。

属性
数据类型

值
字符串

邮件传输用户

电子邮件服务的用户标识。

如果邮件服务器不要求进行认证，那么可以不填写此字段。

属性
数据类型

值
字符串

邮件传输密码

邮件传输用户标识的密码。

如果邮件服务器不要求进行认证，那么可以不填写此字段。

属性	值
数据类型	字符串

业务流程编排器资源管理器 URL

指定业务流程编排器资源管理器的电子邮件链接 URL。

此 URL 用来在生成的电子邮件中提供一个链接，以使接收电子邮件通知的业务管理员可以通过单击该链接在 Web 浏览器中查看相关业务流程或人员任务。

属性	值
数据类型	字符串
示例	http://www.ibm.com:9080/bpc

安全性

在此部分中，指定功能角色到用户标识和组的映射以及业务流程编排器所必需的认证凭证。

管理员用户

管理员安全角色映射到指定的用户标识。

属性	值
数据类型	字符串
缺省值	当前登录用户

管理员组

管理员安全角色映射到指定的组。

属性	值
数据类型	字符串
缺省值	无

监视员用户

系统监视器安全角色映射到指定的用户标识。

属性	值
数据类型	字符串
缺省值	当前登录用户

监视员组

系统监视器安全角色映射到指定的组。

属性	值
数据类型	字符串
缺省值	无

JMS 认证用户

系统集成总线的认证别名。

属性	值
数据类型	字符串
缺省值	当前登录用户

JMS 认证密码和确认密码

JMS 认证用户标识的密码。

属性	值
数据类型	字符串
缺省值	无

JMS API 认证用户

业务流程管理器消息驱动的 Bean 的 RunAs 用户标识。

属性	值
数据类型	字符串
缺省值	当前登录用户

JMS API 认证密码和确认密码

JMS API 认证用户标识的密码。

属性	值
数据类型	字符串
缺省值	无

升级用户认证用户

人员任务管理器消息驱动的 Bean 的 RunAs 用户标识。

属性	值
数据类型	字符串
缺省值	当前登录用户

升级用户认证密码和确认密码

升级用户认证用户标识的密码。

属性	值
数据类型	字符串
缺省值	无

状态观察器

在此部分中，可以对业务流程管理器 and 人员任务管理器启用审计日志记录和公共事件基础结构（CEI）日志记录。

业务流程管理器的审计日志记录

如果选中此项，那么将对业务流程管理器启用审计日志记录。

属性	值
数据类型	复选框
缺省值	选中

人员任务管理器的审计日志记录

如果选中此项，那么将对人员任务管理器启用审计日志记录。

属性	值
数据类型	复选框
缺省值	选中

业务流程管理器的公共事件基础结构日志记录

如果选中此项，那么将对业务流程管理器启用公共事件基础结构日志记录。

属性	值
数据类型	复选框
缺省值	选中

人员任务管理器的公共事件基础结构日志记录

如果选中此项，那么将对人员任务管理器启用公共事件基础结构日志记录。

属性	值
数据类型	复选框
缺省值	选中

SCA 绑定

对于服务组件体系结构（SCA）绑定，可以设置 Web Service API 的上下文根。

主机

这个只读字段显示业务流程管理器绑定和人员任务管理器绑定的主机上下文前缀，上下文根将追加到此前缀。

业务流程管理器的上下文根

业务流程管理器 Web Service 的上下文根。

属性	值
数据类型	字符串
在服务器上配置时的缺省值	/BFMIF_nodeName_serverName

属性	值
在集群中进行配置时的缺省值	/BFMIF_clusterName

人员任务管理器的上下文根

人员任务管理器 Web Service 的上下文根。

属性	值
数据类型	字符串
在服务器上进行配置时的缺省值	/HTMIF_nodeName_serverName
在集群中进行配置时的缺省值	/HTMIF_clusterName

相对路径

这个只读字段显示业务流程管理器和人员任务管理器的 SCA 绑定的相对路径。

属性	值
数据类型	只读字符串
业务流程管理器的相对路径	/sca/com/ibm/bpe/spi/sca/BFMWS
人员任务管理器的相对路径	/sca/com/ibm/task/spi/sca/HTMWS

总线

如果不想让业务流程编排器消息传递引擎使用您为服务组件体系结构 (SCA) 配置的数据源，请展开此部分以更改设置。

使用缺省配置

如果选中此项，那么将使用 SCA 消息传递引擎的当前配置设置。

如果要使用其他设置，那么请不要选中此复选框，以启用此部分中的其他字段。

属性	值
数据类型	复选框
缺省值	选中

总线成员位置

确定是将消息传递引擎的数据存储在本地还是远程位置。

请选择本地或远程。如果选择远程，那么将启用远程目标位置选择器和新建按钮。

属性	值
数据类型	单选按钮
缺省值	本地

远程目标位置

指定远程消息传递引擎存储器的部署目标。

如果此列表为空或者未包含您想要选择的位置，请单击**新建**。

属性	值
数据类型	下拉列表
缺省值	无

新建

此按钮打开“浏览部署目标”页面。

选择部署目标后，该目标将被添加到远程目标位置列表中。

编辑

单击此项以编辑数据源。

测试连接

测试与数据源的连接。

数据库实例

数据库的名称。

属性	值
数据类型	字符串
缺省值	\${USER_INSTALL_ROOT}\databases\BPEME

模式名

要使用的模式的名称。

属性	值
数据类型	字符串
缺省值	MEDBM00

创建表

如果选中此选项，那么第一次访问该数据库时，将自动创建表。要使此选项生效，该数据库必须已存在，并且提供的用户名必须有权在该数据库中创建表和索引。

如果未选中此选项，那么将不会自动创建表，您必须通过运行脚本手动创建这些表。对于生产系统而言，您可能不想使用此选项创建的缺省表。

属性	值
数据类型	复选框
缺省值	选中

用户名

有权连接到数据库和修改数据的用户标识。

如果该用户标识有权在数据库中创建表和索引，那么可以使用用于自动创建表的选项，并且应用服务包或修订包后将根据需要自动更新数据库模式。

属性	值
数据类型	字符串

密码

数据源用户标识的密码。

属性	值
数据类型	字符串

服务器

数据库服务器的地址。

指定主机名/IP 地址以及端口号。

属性	值
数据类型	字符串
示例	localhost:50000

提供程序

业务流程编排器消息传递引擎的 JDBC 提供程序。

如果已将 SCA 配置为使用文件存储器，那么此字段将设置为文件存储器，并且数据库参数字段不可用。选择数据库提供程序后，数据库参数将可用。

属性	值
数据类型	下拉列表
缺省值	您为 SCA 配置的提供程序。

业务流程编排器资源管理器设置

可以在这里设置常规属性。

要查看此管理控制台页面，请单击**服务器** → **集群** → **cluster_name** 或**服务器** → **应用程序服务器** → **server_name**，然后在**业务集成**中展开**业务流程编排器**，接着单击**资源管理器配置**。要创建新的资源管理器配置，请单击**添加**。要查看现有配置，请单击实例名，或者选择实例并单击**编辑**。

上下文根

业务流程编排器资源管理器的上下文根。

此设置定义浏览器访问业务流程编排器资源管理器时必须使用的部分 URL。

属性	值
数据类型	字符串
缺省值	/bpc

资源管理器搜索结果限制

查询返回的最大结果数。

属性	值
数据类型	整型
单位	搜索结果
缺省值	10000

受管业务流程编排器容器

业务流程编排器容器的作用域。

此选项指定此实例可以连接的业务流程编排器容器。这可以是多对一关系。每个业务流程编排器资源管理器实例刚好可以连接至一个业务流程编排器容器。但是，每个业务流程编排器容器可以连接至零个或多个业务流程编排器资源管理器实例。

属性	值
数据类型	部署目标的下拉列表
缺省值	配置了业务流程编排器资源管理器实例的部署目标（服务器或集群）。

业务流程编排器观察器设置

显示业务流程编排器观察器的特定实例的上下文根和监视目标。

要查看此管理控制台页面，请单击服务器 → 集群 → **cluster_name** 或服务器 → 应用程序服务器 → **server_name**，然后在**业务集成**中展开**业务流程编排器**，接着单击**观察器配置**。要创建新的观察器配置，请单击**添加**。要查看现有配置，请单击实例名，或者选择实例并单击**编辑**。

上下文根

业务流程编排器观察器的上下文根。

此设置定义浏览器访问业务流程编排器观察器时必须使用的部分 URL。

属性	值
数据类型	字符串
缺省值	/bpcobserver

对此业务流程编排器事件收集器中的监视数据进行可视化

请选择事件收集器实例，此业务流程编排器观察器实例将对该事件收集器实例的数据进行可视化。

属性	值
数据类型	下拉列表
缺省值	配置了业务流程编排器观察器实例的部署目标（服务器或集群）。

第 4 章 配置业务流程编排器

在安装任何包含业务流程或人员任务的企业应用程序之前，必须对业务流程编排器进行配置。

开始之前

已完成第 81 页的第 3 章，『规划配置业务流程编排器』。

关于此任务

根据所选配置路径的不同，执行下列其中一项任务：

- 对于下列任何非生产配置路径：

- 『基本样本』
- 『公司样本』
- 『非生产部署环境』

请执行『使用安装程序或概要文件管理工具来配置业务流程编排器』。

- 对于『生产部署环境』配置路径，请执行第 127 页的『使用管理控制台的部署环境向导来配置业务流程编排器』。
- 对于『灵活的定制配置』配置路径，根据所要使用的工具的不同，执行下列其中一项操作：
 - 第 129 页的『使用管理控制台的“业务流程编排器”配置页面』
 - 第 140 页的『使用 bpeconfig.jacl 脚本配置业务流程编排器』

结果

业务流程编排器已配置完毕。

下一步做什么？

接着，可以开始对设置进行定制。

使用安装程序或概要文件管理工具来配置业务流程编排器

可以通过五种简单的方法来创建非生产业务流程编排器配置。

开始之前

您已完成第 84 页的『规划创建业务流程编排器基本样本配置』并确定所需的非生产系统样式，如第 83 页的表 4 所述。

过程

1. 根据所选配置路径的不同，执行 1a - 第 126 页的 1c 步骤中的某个步骤。
 - a. 如果要使用未包含用于演示人员分配和替代功能的样本公司的『基本样本』业务流程编排器配置，那么请执行下列操作：
 - 1) 启动安装程序或概要文件管理工具。

- 对于安装程序:
 - 确保选中**典型安装**选项。
 - 确保选中**独立服务器**选项。
 - 确保启用**管理安全性**。
 - 对于概要文件管理工具:
 - 确保创建 **WebSphere Process Server** 概要文件。
 - 确保选中**独立服务器概要文件**选项。
 - 确保选中**典型概要文件创建**选项。
 - 确保选中**启用管理安全性**。
- b. 如果要使用包含用于演示人员分配和替代功能的 15 人样本公司的『公司样本』业务流程编排器配置，那么请执行下列操作:
- 1) 启动概要文件管理工具。
 - 2) 确保创建 **WebSphere Process Server** 概要文件。
 - 3) 确保选中**独立服务器概要文件**选项。
 - 4) 确保选中**高级**选项。
 - 5) 确保选中**根据开发模板创建服务器**选项。
 - 6) 确保选中**启用管理安全性**。
 - 7) 确保选中**配置样本业务流程编排器**。
- c. 如果要使用基于部署环境模式的『非生产部署环境』业务流程编排器配置:
- 1) 启动安装程序或概要文件管理工具。
 - 对于安装程序:
 - 确保选中**部署环境安装**选项。
 - 确保创建 **Deployment Manager**。
 - 可以使业务流程编排器配置基于下列任何模式:
 - 远程消息传递和远程支持
 - 远程消息传递
 - 单个集群
 - 确保启用**管理安全性**，否则您将无法获得业务流程编排器配置。
 - 对于概要文件管理工具:
 - 确保创建 **WebSphere Process Server** 概要文件。
 - 确保选中 **Deployment Manager** 概要文件选项。
 - 可以使业务流程编排器配置基于下列任何模式:
 - 远程消息传递和远程支持
 - 远程消息传递
 - 单个集群
 - 确保启用**管理安全性**，否则您将无法获得业务流程编排器样本。
 - 2) 创建并联合定制概要文件。
2. 可选: 执行第 243 页的『验证业务流程编排器的运行是否正常』。
 3. 可选: 如果要更改 **JMS** 认证用户标识、**RunAs** 用户标识或者角色到用户和组的映射，那么请单击**安全性** → **业务集成安全性**以更改安全性设置。

4. 可选: 更改人员任务管理器的设置:
 - 如果要更改任何与升级电子邮件相关的人员任务管理器设置, 例如发件人地址或者业务流程编排器资源管理器的 URL 前缀, 那么请单击**服务器** → **应用程序服务器** → **server_name** (或者, 如果业务流程编排器被配置在集群中, 那么请单击**服务器** → **集群** → **cluster_name**), 然后在**业务集成**中展开**业务流程编排器容器**, 单击**人员任务管理器**并进行更改。
 - 如果要更改电子邮件服务器地址或端口号或者电子邮件服务器的用户标识或密码, 那么请单击**资源** → **邮件** → **邮件会话**, 选择**单元作用域**, 然后单击 **HTM 邮件会话**并进行更改。
5. 根据用于人员分配功能的人员目录提供程序类型的不同, 可能需要对其进行配置:
 - 在使用系统和用户注册表人员目录提供程序前, 不必对它们进行配置。
 - 如果正在使用轻量级目录访问协议 (LDAP), 那么请执行第 175 页的『配置 LDAP 人员目录提供程序』。
 - 如果正在使用虚拟成员管理器 (VMM), 那么请执行第 173 页的『配置虚拟成员管理器人员目录提供程序』。
6. 可选: 如果已配置 VMM, 并且要使用人员替代功能, 那么请执行第 180 页的『配置人员替代功能』。
7. 可选: 如果要使用组工作项, 那么请使用管理控制台来启用这些项。单击**服务器** → **应用程序服务器** → **server_name** (或者, 如果业务流程编排器被配置在集群中, 那么请单击**服务器** → **集群** → **cluster_name**), 接着在**业务集成**中展开**业务流程编排器容器**并单击**人员任务管理器**, 然后选中**启用组工作项**。

结果

业务流程编排器已配置完毕。

使用管理控制台的部署环境向导来配置业务流程编排器

通过使用管理控制台的部署环境向导, 可以创建基于模式并包含业务流程编排器的配置。如果业务流程编排器配置有自己的数据库, 那么此配置适用于生产系统。

开始之前

您已执行第 86 页的『规划使用管理控制台的部署环境向导』。

过程

1. 启动部署环境向导。在管理控制台中, 单击**服务器** → **部署环境** → **新建**。在输入其他配置参数时, 请确保输入第 86 页的『规划使用管理控制台的部署环境向导』中规划的值:
 - a. 可以使业务流程编排器配置基于下列任何模式:
 - 远程消息传递和远程支持
 - 远程消息传递
 - 单个集群
 - 定制
 - b. 在“安全性”页面中, 可以设置用户名和密码, 此用户名和密码将用作业务流程编排器 (标识为 WBI_BPC 组件) 的认证别名。

- c. 在数据库页面中，如果要让业务流程编排器、业务流程编排器资源管理器或业务流程编排器消息传递引擎使用不同的数据库，那么请将缺省数据源由缺省值更改为规划的值。
 - d. 在“业务流程编排器”页面中，指定为此配置规划的上下文根、安全性参数和邮件会话参数。
 2. 如果对业务流程编排器指定了独立的数据库，那么请执行第 162 页的『使用 SQL 脚本为业务流程编排器创建数据库』。否则，对于非 Derby 数据库，请确保存在空数据库，以使业务流程编排器在第一次访问该数据库时能够在该数据库中创建缺省模式。
 3. 如果对业务流程编排器观察器指定了独立的数据库，那么请执行第 193 页的『为业务流程编排器观察器准备数据库』。否则，对于非 Derby 数据库，请确保存在空数据库，以使业务流程编排器在第一次访问该数据库时能够在该数据库中创建缺省模式。
 4. 如果对业务流程编排器消息传递引擎指定了独立的数据库，那么请确保该数据库存在。
 - 如果要使用**创建表**选项，以便让消息传递引擎在第一次使用该数据库时创建缺省模式，那么请授权数据库用户标识在您计划使用的模式中创建表和视图。
 - 否则，如果您计划**不使用创建表**选项，那么请在缺省消息传递提供程序尝试访问该数据库之前创建表。可以使用 *install_root* 目录中 bin 子目录中的 sibDDLGenerator 实用程序来生成可用于创建表的 DDL 文件。
 5. 对于每个配置了业务流程编排器的节点，确保设置 JDBC 驱动程序的环境变量。在集群中，必须对每个主管集群成员的节点执行此操作。
 - a. 单击**环境** → **WebSphere 变量**，对于**作用域**，请选择配置了业务流程编排器的节点。
 - b. 选择 JDBC 提供程序的环境变量：
 - 对于 Derby，不需要设置任何环境变量。
 - 对于 DB2 Linux 版、UNIX 版、Windows 版或 z/OS 版，如果正在使用 CLI 驱动程序，那么请选择 DB2_JDBC_DRIVER_PATH。
 - 对于 DB2 Linux 版、UNIX 版、Windows 版或 z/OS 版，如果正在使用通用驱动程序，那么请选择 DB2UNIVERSAL_JDBC_DRIVER_PATH。
 - 对于 DB2 i5/OS 版，如果正在使用本机驱动程序，那么请选择 OS400_NATIVE_JDBC_DRIVER_PATH。
 - 对于 DB2 i5/OS 版，如果正在使用 Toolbox 驱动程序，那么请选择 OS400_TOOLBOX_JDBC_DRIVER_PATH。
 - 对于 Oracle，请选择 ORACLE_JDBC_DRIVER_PATH。
 - 对于 Informix，请选择 INFORMIX_JDBC_DRIVER_PATH。
 - 对于使用 WebSphere 嵌入式 ConnectJDBC 驱动程序的 SQL Server，不需要设置任何环境变量。
 - 对于使用 DataDirect ConnectJDBC 4 类驱动程序的 SQL Server，请选择 CONNECTJDBC_JDBC_DRIVER_PATH。
 - c. 将此环境变量设置为指向 JDBC 驱动程序的 JAR 文件位置。
 6. 激活业务流程编排器： 执行第 242 页的『激活业务流程编排器』。
 7. 可选： 验证业务流程编排器基本配置能否正常运行： 执行第 243 页的『验证业务流程编排器的运行是否正常』。

8. 可选: 更改人员任务管理器的设置:
 - 如果要更改任何与升级电子邮件相关的人员任务管理器设置, 例如发件人地址或者业务流程编排器资源管理器的 URL 前缀, 那么请单击**服务器** → **应用程序服务器** → **server_name** (或者, 如果业务流程编排器被配置在集群中, 那么请单击**服务器** → **集群** → **cluster_name**), 然后在**业务集成**中展开**业务流程编排器容器**, 单击**人员任务管理器**并进行更改。
 - 如果要更改电子邮件服务器地址或端口号或者电子邮件服务器的用户标识或密码, 那么请单击**资源** → **邮件** → **邮件会话**, 选择**单元**作用域, 然后单击 **HTM 邮件会话**并进行更改。
9. 根据用于人员分配功能的人员目录提供程序类型的不同, 可能需要对其进行配置:
 - 在使用系统和用户注册表人员目录提供程序前, 不必对它们进行配置。
 - 如果正在使用轻量级目录访问协议 (LDAP), 那么请执行第 175 页的『配置 LDAP 人员目录提供程序』。
 - 如果正在使用虚拟成员管理器 (VMM), 那么请执行第 173 页的『配置虚拟成员管理器人员目录提供程序』。
10. 可选: 如果已配置 VMM, 并且要使用人员替代功能, 那么请执行第 180 页的『配置人员替代功能』。
11. 可选: 如果要使用组工作项, 那么请使用管理控制台来启用这些项。单击**服务器** → **应用程序服务器** → **server_name** (或者, 如果业务流程编排器被配置在集群中, 那么请单击**服务器** → **集群** → **cluster_name**), 接着在**业务集成**中展开**业务流程编排器容器**并单击**人员任务管理器**, 然后选中**启用组工作项**。
12. 如果已启用应用程序安全性, 并且长时间运行流程调用远程 EJB 方法, 那么请确保公共安全互操作性 V2 (CSIv2) 入站认证配置启用 CSIv2 身份声明。

结果

业务流程编排器已针对所选部署环境配置完毕。

使用管理控制台的“业务流程编排器”配置页面

本节描述如何使用管理控制台的“业务流程编排器”配置页面在给定的服务器或集群中创建配置。

关于此任务

在运行包含业务流程或人员任务的应用程序前, 必须先配置必需的资源并安装业务流程编排器应用程序。

过程

1. 如果创建缺省概要文件时选择了业务流程编排器样本配置选项, 那么业务流程管理器、人员任务管理器、业务流程编排器资源管理器和业务流程编排器观察器均已配置完毕。

通过在管理控制台中查找名称以下列各项开头的企业应用程序, 可以检查是否已配置这些应用程序:

- BPCObserver

- BPCECollector
- BPEContainer
- BPCEExplorer
- TaskContainer

样本配置使用 Derby 数据库，并且不适用于生产系统。由于在部署目标上只能有一个业务流程编排器配置，所以必须先按第 247 页的第 5 章，『除去业务流程编排器配置』中描述的方法除去样本配置，然后才能继续配置业务流程编排器。

- 在 Network Deployment 环境中，请确保配置服务组件体系结构 (SCA)：
 - 如果要在服务器上配置业务流程编排器，那么请单击**服务器** → **应用程序服务器** → **serverName**，然后在**业务集成**部分中单击**服务组件体系结构**。
 - 如果要在集群中配置业务流程编排器，那么请单击**服务器** → **集群** → **clusterName**，然后在**业务集成**部分中单击**服务组件体系结构**。
 - 如果尚未启用**支持服务组件体系结构组件**，那么请将其选中。
- 为业务流程编排器消息传递引擎的数据存储器创建数据库：
 - 如果要使用“业务流程编排器”配置页面上的**创建表**选项，以便让消息传递引擎在第一次使用该数据库时创建缺省模式，那么请执行下列操作：
 - 如果该数据库尚未存在，那么请进行创建。
 - 授权数据库用户标识在您计划使用的模式中创建表和视图。
 - 否则，如果您计划不使用**创建表**选项，那么请在缺省消息传递提供程序尝试访问该数据库之前创建表。可以使用 *install_root* 目录中 bin 子目录中的 sibDDLGenerator 实用程序来生成可用于创建表的 DDL 文件。
- 如果正在使用 WebSphere MQ 作为 Java 消息服务 (JMS) 提供程序，那么必须创建队列管理器和队列：请执行第 183 页的『创建业务流程编排器的队列管理器和队列』。如果正在使用缺省消息传递 JMS 提供程序，那么无需执行此步骤。
- 为业务流程编排器创建数据库：
 - 如果要使用“业务流程编排器”配置页面上的**创建表**选项，以便让业务流程编排器在第一次使用该数据库时创建缺省模式并在用户的缺省表空间中创建所有的表，那么请执行下列操作：
 - 如果该数据库尚未存在，那么请进行创建。
 - 授权数据库用户标识在您计划使用的模式中创建表和视图。
 - 否则，如果计划不使用**创建表**选项，那么请执行第 162 页的『使用 SQL 脚本为业务流程编排器创建数据库』。
- 对于每个配置了业务流程编排器的节点，确保设置 JDBC 驱动程序的环境变量。在集群中，必须对每个主管集群成员的节点执行此操作。
 - 单击**环境** → **WebSphere 变量**，对于**作用域**，请选择配置了业务流程编排器的节点。
 - 选择 JDBC 提供程序的环境变量：
 - 对于 Derby，不需要设置任何环境变量。
 - 对于 DB2 Linux 版、UNIX 版、Windows 版或 z/OS 版，如果正在使用 CLI 驱动程序，那么请选择 DB2_JDBC_DRIVER_PATH。
 - 对于 DB2 Linux 版、UNIX 版、Windows 版或 z/OS 版，如果正在使用通用驱动程序，那么请选择 DB2UNIVERSAL_JDBC_DRIVER_PATH。

- 对于 DB2 i5/OS 版，如果正在使用本机驱动程序，那么请选择 OS400_NATIVE_JDBC_DRIVER_PATH。
 - 对于 DB2 i5/OS 版，如果正在使用 Toolbox 驱动程序，那么请选择 OS400_TOOLBOX_JDBC_DRIVER_PATH。
 - 对于 Oracle，请选择 ORACLE_JDBC_DRIVER_PATH。
 - 对于 Informix，请选择 INFORMIX_JDBC_DRIVER_PATH。
 - 对于使用 WebSphere 嵌入式 ConnectJDBC 驱动程序的 SQL Server，不需要设置任何环境变量。
 - 对于使用 DataDirect ConnectJDBC 4 类驱动程序的 SQL Server，请选择 CONNECTJDBC_JDBC_DRIVER_PATH。
- c. 将此环境变量设置为指向 JDBC 驱动程序的 JAR 文件位置。
7. 在管理控制台中，选择要配置业务流程编排器的服务器或集群。单击下列其中一项：
- 服务器 → 应用程序服务器 → *serverName*
 - 服务器 → 集群 → *clusterName*

其中，*serverName* 或 *clusterName* 是服务器或集群的名称。

8. 转到“业务流程编排器”配置页面：在容器设置部分中，展开业务流程编排器容器设置，然后单击业务流程编排器容器。
9. 验证是否未配置业务流程编排器。应该会有一条消息指示当前未安装业务流程编排器容器（业务流程管理器和人员任务管理器）。

如果已安装业务流程管理器和人员任务管理器，那么请执行第 247 页的第 5 章，『除去业务流程编排器配置』，然后再转到下一个步骤。

10. 对于此服务器或集群中的业务流程编排器配置，请输入值并选择要规划的选项。有关更多信息，请参阅第 115 页的『业务流程编排器配置』，此配置分为下列部分：
- a. 第 116 页的『数据源』
 - b. 第 117 页的『人员任务管理器邮件会话』
 - c. 第 118 页的『安全性』
 - d. 第 120 页的『状态观察器』
 - e. 第 120 页的『SCA 绑定』
 - f. 第 121 页的『总线』
11. 单击 **Apply**。将显示信息，以报告业务流程编排器的部署和配置进度。
12. 如果安装成功，请单击 **保存主配置**，然后单击 **保存**。否则，废弃更改，接着在管理控制台以及 Deployment Manager 或服务器上的 SystemOut.log 文件中查找错误消息，这有助于解决问题。然后，再次进行尝试。
13. 激活业务流程编排器：执行第 242 页的『激活业务流程编排器』。
14. 可选：验证业务流程编排器基本配置能否正常运行：执行第 243 页的『验证业务流程编排器的运行是否正常』。
15. 可选：更改人员任务管理器的设置：
- 如果要更改任何与升级电子邮件相关的人员任务管理器设置，例如发件人地址或者业务流程编排器资源管理器的 URL 前缀，那么请单击 **服务器** → **应用程序服**

- 务器 → **server_name** (或者, 如果业务流程编排器被配置在集群中, 那么请单击服务器 → 集群 → **cluster_name**), 然后在业务集成中展开业务流程编排器容器, 单击人员任务管理器并进行更改。
- 如果要更改电子邮件服务器地址或端口号或者电子邮件服务器的用户标识或密码, 那么请单击资源 → 邮件 → 邮件会话, 选择单元作用域, 然后单击 **HTM 邮件会话** 并进行更改。
16. 根据用于人员分配功能的人员目录提供程序类型的不同, 可能需要对其进行配置:
 - 在使用系统和用户注册表人员目录提供程序前, 不必对它们进行配置。
 - 如果正在使用轻量级目录访问协议 (LDAP), 那么请执行第 175 页的『配置 LDAP 人员目录提供程序』。
 - 如果正在使用虚拟成员管理器 (VMM), 那么请执行第 173 页的『配置虚拟成员管理器人员目录提供程序』。
 17. 可选: 如果已配置 VMM, 并且要使用人员替代功能, 那么请执行第 180 页的『配置人员替代功能』。
 18. 可选: 如果要使用组工作项, 那么请使用管理控制台来启用这些项。单击服务器 → 应用程序服务器 → **server_name** (或者, 如果业务流程编排器被配置在集群中, 那么请单击服务器 → 集群 → **cluster_name**), 接着在业务集成中展开业务流程编排器容器并单击人员任务管理器, 然后选中启用组工作项。
 19. 如果已启用应用程序安全性, 并且长时间运行流程调用远程 EJB 方法, 那么请确保公共安全互操作性 V2 (CSIv2) 入站认证配置启用 CSIv2 身份声明。
 20. 可选: 如果尚未安装和配置业务流程编排器资源管理器, 那么现在可以对其进行配置。请执行第 188 页的『配置业务流程编排器资源管理器』。
 21. 可选: 如果尚未安装和配置业务流程编排器观察器, 那么现在可以对其进行配置。请执行第 191 页的『配置业务流程编排器观察器』。

结果

业务流程编排器已配置完毕。

业务流程编排器配置

使用此面板来安装和配置业务流程编排器。

要查看此管理控制台页面, 请单击服务器 → 集群 → **cluster_name**或服务器 → 应用程序服务器 → **server_name**, 然后在容器中单击业务流程编排器容器。

此页面分为多个部分。有关每一部分中各个字段的详细信息, 请参阅:

- 第 116 页的『数据源』
- 第 117 页的『人员任务管理器邮件会话』
- 第 118 页的『安全性』
- 第 120 页的『状态观察器』
- 第 120 页的『SCA 绑定』
- 第 121 页的『总线』

数据源

在此部分中，指定业务流程编排器的数据源。

编辑

单击此项以编辑数据源。

测试连接

测试与数据源的连接。

数据库实例

业务流程管理器和人员任务管理器使用的数据库的名称。

属性	值
数据类型	字符串
缺省值	WPRCSDB

模式名

要使用的模式的名称。

仅当您想使用自己的模式来代替缺省模式时，才需要更改模式名。

属性	值
数据类型	字符串

创建表

如果选中此选项，那么第一次访问该数据库时，将自动创建表。要使此选项生效，该数据库必须已存在，并且提供的用户名必须有权在该数据库中创建表和索引。

对于生产系统而言，建议您不要使用此选项。如果未选中此选项，那么将不会自动创建表，您必须通过运行脚本手动创建这些表。

属性	值
数据类型	复选框
缺省值	选中

用户名

有权连接到数据库和修改数据的用户标识。

如果该用户标识有权在数据库中创建表和索引，那么可以使用用于自动创建表的选项，并且应用服务包或修订包后将根据需要自动更新数据库模式。

属性	值
数据类型	字符串

密码

数据源用户标识的密码。

属性	值
数据类型	字符串

服务器

数据库服务器的地址。

指定主机名/IP 地址以及端口号。

属性	值
数据类型	字符串
示例	localhost:50000

提供程序

业务流程编排器的 JDBC 提供程序。

属性	值
数据类型	下拉列表

人员任务管理器邮件会话

在此部分中，指定升级电子邮件的参数。

启用电子邮件服务

如果人员任务管理器将发送升级电子邮件通知，那么必须启用邮件会话。

属性	值
数据类型	复选框
缺省值	选中

邮件传输主机

简单电子邮件传输协议（SMTP）电子邮件服务的主机名或 IP 地址。

属性	值
数据类型	字符串

邮件传输用户

电子邮件服务的用户标识。

如果邮件服务器不要求进行认证，那么可以不填写此字段。

属性	值
数据类型	字符串

邮件传输密码

邮件传输用户标识的密码。

如果邮件服务器不要求进行认证，那么可以不填写此字段。

属性	值
数据类型	字符串

业务流程编排器资源管理器 URL

指定业务流程编排器资源管理器的电子邮件链接 URL。

此 URL 用来在生成的电子邮件中提供一个链接，以使接收电子邮件通知的业务管理员可以通过单击该链接在 Web 浏览器中查看相关业务流程或人员任务。

属性	值
数据类型	字符串
示例	http://www.ibm.com:9080/bpc

安全性

在此部分中，指定功能角色到用户标识和组的映射以及业务流程编排器所必需的认证凭证。

管理员用户

管理员安全角色映射到指定的用户标识。

属性	值
数据类型	字符串
缺省值	当前登录用户

管理员组

管理员安全角色映射到指定的组。

属性	值
数据类型	字符串
缺省值	无

监视员用户

系统监视器安全角色映射到指定的用户标识。

属性	值
数据类型	字符串
缺省值	当前登录用户

监视员组

系统监视器安全角色映射到指定的组。

属性	值
数据类型	字符串
缺省值	无

JMS 认证用户

系统集成总线的认证别名。

属性	值
数据类型	字符串
缺省值	当前登录用户

JMS 认证密码和确认密码

JMS 认证用户标识的密码。

属性	值
数据类型	字符串
缺省值	无

JMS API 认证用户

业务流程管理器消息驱动的 Bean 的 RunAs 用户标识。

属性	值
数据类型	字符串
缺省值	当前登录用户

JMS API 认证密码和确认密码

JMS API 认证用户标识的密码。

属性	值
数据类型	字符串
缺省值	无

升级用户认证用户

人员任务管理器消息驱动的 Bean 的 RunAs 用户标识。

属性	值
数据类型	字符串
缺省值	当前登录用户

升级用户认证密码和确认密码

升级用户认证用户标识的密码。

属性	值
数据类型	字符串
缺省值	无

状态观察器

在此部分中，可以对业务流程管理器和人员任务管理器启用审计日志记录和公共事件基础结构（CEI）日志记录。

业务流程管理器的审计日志记录

如果选中此项，那么将对业务流程管理器启用审计日志记录。

属性	值
数据类型	复选框
缺省值	选中

人员任务管理器的审计日志记录

如果选中此项，那么将对人员任务管理器启用审计日志记录。

属性	值
数据类型	复选框
缺省值	选中

业务流程管理器的公共事件基础结构日志记录

如果选中此项，那么将对业务流程管理器启用公共事件基础结构日志记录。

属性	值
数据类型	复选框
缺省值	选中

人员任务管理器的公共事件基础结构日志记录

如果选中此项，那么将对人员任务管理器启用公共事件基础结构日志记录。

属性	值
数据类型	复选框
缺省值	选中

SCA 绑定

对于服务组件体系结构 (SCA) 绑定，可以设置 Web Service API 的上下文根。

主机

这个只读字段显示业务流程管理器绑定和人员任务管理器绑定的主机上下文前缀，上下文根将追加到此前缀。

业务流程管理器的上下文根

业务流程管理器 Web Service 的上下文根。

属性	值
数据类型	字符串
在服务器上进行配置时的缺省值	<code>/BFMIF_nodeName_serverName</code>
在集群中进行配置时的缺省值	<code>/BFMIF_clusterName</code>

人员任务管理器的上下文根

人员任务管理器 Web Service 的上下文根。

属性	值
数据类型	字符串

属性	值
在服务器上进行配置时的缺省值	/HTMIF_nodeName_serverName
在集群中进行配置时的缺省值	/HTMIF_clusterName

相对路径

这个只读字段显示业务流程管理器 and 人员任务管理器的 SCA 绑定的相对路径。

属性	值
数据类型	只读字符串
业务流程管理器的相对路径	/sca/com/ibm/bpe/spi/sca/BFMWS
人员任务管理器的相对路径	/sca/com/ibm/task/spi/sca/HTMWS

总线

如果不想让业务流程编排器消息传递引擎使用您为服务组件体系结构 (SCA) 配置的数据源，请展开此部分以更改设置。

使用缺省配置

如果选中此项，那么将使用 SCA 消息传递引擎的当前配置设置。

如果要使用其他设置，那么请不要选中此复选框，以启用此部分中的其他字段。

属性	值
数据类型	复选框
缺省值	选中

总线成员位置

确定是将消息传递引擎的数据存储在本地还是远程位置。

请选择本地或远程。如果选择远程，那么将启用远程目标位置选择器和新建按钮。

属性	值
数据类型	单选按钮
缺省值	本地

远程目标位置

指定远程消息传递引擎存储器的部署目标。

如果此列表为空或者未包含您想要选择的位置，请单击新建。

属性	值
数据类型	下拉列表
缺省值	无

新建

此按钮打开“浏览部署目标”页面。

选择部署目标后，该目标将被添加到远程目标位置列表中。

编辑

单击此项以编辑数据源。

测试连接

测试与数据源的连接。

数据库实例

数据库的名称。

属性	值
数据类型	字符串
缺省值	\${USER_INSTALL_ROOT}\databases\BPEME

模式名

要使用的模式的名称。

属性	值
数据类型	字符串
缺省值	MEDBM00

创建表

如果选中此选项，那么第一次访问该数据库时，将自动创建表。要使此选项生效，该数据库必须已存在，并且提供的用户名必须有权在该数据库中创建表和索引。

如果未选中此选项，那么将不会自动创建表，您必须通过运行脚本手动创建这些表。对于生产系统而言，您可能不想使用此选项创建的缺省表。

属性	值
数据类型	复选框
缺省值	选中

用户名

有权连接到数据库和修改数据的用户标识。

如果该用户标识有权在数据库中创建表和索引，那么可以使用用于自动创建表的选项，并且应用服务包或修订包后将根据需要自动更新数据库模式。

属性	值
数据类型	字符串

密码

数据源用户标识的密码。

属性	值
数据类型	字符串

服务器

数据库服务器的地址。

指定主机名/IP 地址以及端口号。

属性	值
数据类型	字符串
示例	localhost:50000

提供程序

业务流程编排器消息传递引擎的 JDBC 提供程序。

如果已将 SCA 配置为使用文件存储器，那么此字段将设置为文件存储器，并且数据库参数字段不可用。选择数据库提供程序后，数据库参数将可用。

属性	值
数据类型	下拉列表
缺省值	您为 SCA 配置的提供程序。

使用 bpeconfig.jacl 脚本配置业务流程编排器

本节描述如何使用 bpeconfig.jacl 脚本在给定的服务器或集群中配置业务流程编排器以及所有必需资源。

过程

1. 确保您了解将要使用的选项和参数。请参阅第 81 页的第 3 章，『规划配置业务流程编排器』中规划的值。必须在批处理文件或命令行中指定所有必需参数和选项。否则，系统将提示您输入未提供的必需参数。有关此脚本及其选项和参数的详细信息，请参阅第 146 页的『bpeconfig.jacl 脚本文件』。

选项	描述
如果服务器或者 Network Deployment 环境中的 Deployment Manager 未处于运行状态	使用以下选项： -conntype NONE 如果服务器或 Deployment Manager 处于运行状态，那么请不要使用此选项。
如果已启用管理安全性	指定下列参数： -user <i>userName</i> -password <i>userPassword</i>
如果未使用缺省概要文件	指定以下参数： -profileName <i>profileName</i>
如果不是在缺省服务器上配置业务流程编排器	指定以下参数： -cluster <i>clusterName</i> 或同时指定这两个参数： -node <i>nodeName</i> -server <i>serverName</i>

选项	描述
由于此脚本始终创建业务流程编排器配置	<p>指定业务流程管理器和人员任务管理器所需的参数:</p> <pre> {-adminBFMUsers userList -adminBFMGroups groupList} {-monitorBFMUsers userList -monitorBFMGroups groupList} -jmsBFMRunAsUser userID -jmsBFMRunAsPwd password {-adminHTMUsers userList -adminHTMGroups groupList} {-monitorHTMUsers userList -monitorHTMGroups groupList} -jmsHTMRunAsUser userID -jmsHTMRunAsPwd password -contextRootBFM contextRootBFM -contextRootHTM contextRootHTM </pre> <p>对于以 <i>Users</i> 和 <i>Groups</i> 结尾的参数对, 必须指定这一对参数中的一个或全部。两个以 <i>contextRoot</i> 开头的参数是可选的。</p>
如果要启用简单电子邮件传输协议 (SMTP) 服务器以发送升级电子邮件	<p>指定以下参数:</p> <pre>-mailServerName mailServerName</pre> <p>如果邮件服务器要求进行认证, 那么还应该指定下列参数:</p> <pre>-mailUser mailUserID -mailPwd mailPassword</pre>

选项	描述
<p>由于可以让脚本文件创建数据库，也可以让它仅仅生成 SQL 脚本而不运行那些脚本</p>	<p>使用以下选项：</p> <pre data-bbox="935 264 1198 296">-createDB { yes no }</pre> <p>如果选择 yes，那么 <code>bpeconfig.jacl</code> 脚本将生成并运行用于在缺省表空间中创建数据库对象的 SQL 文件，这不适合于高性能系统。在这种情况下，还需要规划停止服务器并指定 <code>-conntype NONE</code> 选项。</p> <p>如果选择 no，并且该数据库尚未存在，那么您或数据库管理员必须运行所生成的 SQL 脚本。对于高性能系统而言，由于运行 SQL 脚本前需要对其进行定制，因此请指定 no。如果您无权自己创建数据库，那么也需指定 no，以便能够将该 SQL 脚本提供给数据库管理员进行定制和运行。</p> <p>如果正在使用支持受限制的数据库，那么也必须指定 no。</p> <p>限制： 此脚本无法创建下列类型的数据库：</p> <ul data-bbox="935 894 1281 1052" style="list-style-type: none"> • DB2 z/OS 版 • Oracle • 远程 Microsoft SQL Server • 远程 Informix Dynamic Server <p>如果选择 yes 并以连接方式运行此脚本，那么在创建数据库或模式时，如果运行时间超过缺省超时值（3 分钟），此操作将失败。</p>

选项	描述
由于每个业务流程编排器配置都需要访问数据库	<p>指定以下参数:</p> <pre>-dbType <i>databaseType</i></pre> <p>并且, 请提供数据库类型所需的参数 (有关详细信息, 请参阅第 146 页的『bpeconfig.jacl 脚本文件』):</p> <pre>-dbVersion <i>version</i> -dbHome <i>databaseInstallPath</i> -dbJava <i>JDBCdriverPath</i> -dbName <i>databaseName</i> -dbUser <i>databaseUser</i> -dbPwd <i>databasePassword</i> -dbAdmin <i>databaseAdministratorUserID</i> -driverType <i>JDBCdriverType</i> -dbTablespaceDir <i>databaseTablespacePath</i> -dbServerName <i>databaseServerName</i> -dbServerPort <i>databaseServerPort</i> -dbStorageGroup <i>DB2zOSSStorageGroup</i> -dbConnectionTarget <i>DB2zOSSSubSystem</i> -dbSchema <i>schemaQualifier</i> -dbInstance <i>InformixInstance</i></pre> <p>在集群中以批处理方式运行此脚本时, 如果数据库要求指定 <code>-dbJava</code> 参数, 那么请按以下方式对每个主管集群成员的节点指定此参数:</p> <pre>-dbJava.<i>nodeName</i> <i>JDBCdriverPath</i> _on_<i>nodeName</i></pre> <p>注: 如果正在使用下列其中一种数据库, 那么 <code>bpeconfig.jacl</code> 还可以创建数据库实例:</p> <ul style="list-style-type: none"> • 本地 DB2 Linux 版、UNIX 版或 Windows 版 • DB2 iSeries 版 • 嵌入式 Derby • Derby Network 数据库, 并且服务器处于运行状态
由于每个业务流程编排器配置都使用 JMS 提供程序	<p>指定以下参数:</p> <pre>-mqType { WPM MQSeries }</pre> <p>并且, 请提供 JMS 提供程序所需的参数 (有关详细信息, 请参阅第 146 页的『bpeconfig.jacl 脚本文件』):</p> <pre>-createQM { yes no } -qmNameGet <i>getQueueManagerName</i> -mqClusterName <i>mqClusterName</i> -qmNamePut <i>putQueueManagerName</i> -mqHome <i>MQInstallationDirectory</i> -mqUser <i>JMSProviderUserID</i> -mqPwd <i>JMSProviderPassword</i></pre> <p>注: 不推荐使用 MQSeries® 选项。</p>

选项	描述
如果使用了 <code>-mqType WPM</code> 选项，那么请指定消息传递引擎存储库设置	指定下列参数： <pre>-meStoreType { FILESTORE DATASTORE } -mqCreateTables { true false } -mqSchemaName mqSchemaName -mqDataSource datasourceName -medbUser meDatabaseUser -medbPwd meDatabasePassword</pre>
由于此脚本始终配置业务流程编排器资源管理器	指定下列任何可选参数： <pre>-hostName explorerVirtualHostname -explorerHost explorerURL -remoteNode nodeName -remoteServer serverName -remoteCluster clusterName -contextRootExplorer explorerContextRoot -maxListEntries maximum</pre> <p>有关这些参数的更多信息，包括缺省值，请参阅第 189 页的『使用 <code>clientconfig.jacl</code> 脚本文件配置业务流程编排器资源管理器』。</p>
了解是否要在部署目标上安装并配置业务流程编排器观察器或事件收集器应用程序	使用下列选项： <pre>-createEventCollector { yes no } -createObserver { yes no }</pre> <p>仅当以批处理方式运行 <code>bpeconfig.jacl</code> 时，才能使用这些选项；并且，它们不适用于高性能系统。对于生产系统，请执行第 191 页的『配置业务流程编排器观察器』。</p>

- 如果创建缺省概要文件时选择了业务流程编排器样本配置选项，那么业务流程管理器、人员任务管理器、业务流程编排器资源管理器和业务流程编排器观察器均已配置完毕。

通过在管理控制台中查找名称以下列各项开头的企业应用程序，可以检查是否已配置这些应用程序：

- BPCObserver
- BPCECollector
- BPEContainer
- BPCEplorer
- TaskContainer

样本配置使用 Derby 数据库，并且不适用于生产系统。由于在部署目标上只能有一个业务流程编排器配置，所以必须先按第 247 页的第 5 章，『除去业务流程编排器配置』中描述的方法除去样本配置，然后才能继续配置业务流程编排器。

- 在 Network Deployment 环境中，请确保配置服务组件体系结构 (SCA)：
 - 如果要在服务器上配置业务流程编排器，那么请单击服务器 → 应用程序服务器 → **serverName**，然后在业务集成部分中单击服务组件体系结构。
 - 如果要在集群中配置业务流程编排器，那么请单击服务器 → 集群 → **clusterName**，然后在业务集成部分中单击服务组件体系结构。
 - 如果尚未启用支持服务组件体系结构组件，那么请将其选中。

4. 如果正在使用 WebSphere 平台消息传递 (WPM) 作为 JMS 提供程序, 并且未使用选项 `-meStoreType DATASTORE` (对于嵌入式 Derby 数据库) 或 `-meStoreType FILESTORE`, 那么请为业务流程编排器消息传递引擎的数据存储器创建数据库:
 - 如果要使用 `-mqCreateTables yes` 选项, 以便让消息传递引擎在它第一次使用数据库时创建缺省模式, 那么请执行下列操作:
 - a. 如果该数据库尚未存在, 那么请进行创建。
 - b. 授权数据库用户标识在您计划使用的模式中创建表和视图。
 - 否则, 如果将要使用 `-mqCreateTables no` 选项, 那么请在缺省消息传递提供程序尝试访问该数据库之前创建表。可以使用 `install_root` 目录中 `bin` 子目录中的 `sibDDLGenerator` 实用程序来生成可用于创建表的 DDL 文件。
5. 如果计划使用选项 `-createDB yes` 来运行所生成的 SQL 脚本以便创建数据库模式, 那么请执行下列操作:
 - a. 如果正在使用下列其中一种数据库:
 - DB2 z/OS 版
 - Oracle
 - 远程 Microsoft SQL Server
 - 远程 Informix Dynamic Server并且该数据库尚未存在, 那么请参照数据库文档手动创建空数据库。
 - b. 确保数据库客户机 (例如 `db2.exe`) 在脚本编制客户机的路径中。
 - c. 确保应用程序服务器已停止。
6. 以批处理方式 (需提供已规划的选项和配置参数) 或交互方式调用 `bpeconfig.jacl` 脚本文件。有关此脚本文件的详细信息, 请参阅第 146 页的『`bpeconfig.jacl` 脚本文件』。
7. 如果已使用 `-createDB no` 选项来推迟创建数据库, 或者 `bpeconfig.jacl` 脚本未能创建该数据库, 那么在步骤 8 中激活业务流程编排器之前, 您或数据库管理员应该执行第 159 页的『使用生成的 SQL 脚本为业务流程编排器创建数据库模式』中描述的操作。

注: 如果正在使用本地数据库并且您在步骤 8 中激活业务流程编排器前已存在该数据库, 并且您未执行第 159 页的『使用生成的 SQL 脚本为业务流程编排器创建数据库模式』中描述的操作, 那么业务流程编排器第一次访问该数据库时, 将创建缺省模式。

8. 激活业务流程编排器: 执行第 242 页的『激活业务流程编排器』。
9. 可选: 验证业务流程编排器基本配置能否正常运行: 执行第 243 页的『验证业务流程编排器的运行是否正常』。
10. 可选: 如果要更改 JMS 认证用户标识、RunAs 用户标识或者角色到用户和组的映射, 那么请单击安全性 → 业务集成安全性以更改安全性设置。
11. 可选: 更改人员任务管理器的设置:
 - 如果要更改任何与升级电子邮件相关的人员任务管理器设置, 例如发件人地址或者业务流程编排器资源管理器的 URL 前缀, 那么请单击服务器 → 应用程序服务器 → `server_name` (或者, 如果业务流程编排器被配置在集群中, 那么请单击服务器 → 集群 → `cluster_name`), 然后在业务集成中展开业务流程编排器容器, 单击人员任务管理器并进行更改。

- 如果要更改电子邮件服务器地址或端口号或者电子邮件服务器的用户标识或密码，那么请单击资源 → 邮件 → 邮件会话，选择单元作用域，然后单击 **HTM 邮件会话** 并进行更改。
12. 根据用于人员分配功能的人员目录提供程序类型的不同，可能需要对其进行配置：
 - 在使用系统和用户注册表人员目录提供程序前，不必对它们进行配置。
 - 如果正在使用轻量级目录访问协议（LDAP），那么请执行第 175 页的『配置 LDAP 人员目录提供程序』。
 - 如果正在使用虚拟成员管理器（VMM），那么请执行第 173 页的『配置虚拟成员管理器人员目录提供程序』。
 13. 可选：如果已配置 VMM，并且要使用人员替代功能，那么请执行第 180 页的『配置人员替代功能』。
 14. 可选：如果要使用组工作项，那么请使用管理控制台来启用这些项。单击**服务器** → **应用程序服务器** → **server_name**（或者，如果业务流程编排器被配置在集群中，那么请单击**服务器** → **集群** → **cluster_name**），接着在**业务集成**中展开**业务流程编排器容器**并单击**人员任务管理器**，然后选中**启用组工作项**。
 15. 如果已启用应用程序安全性，并且长时间运行流程调用远程 EJB 方法，那么请确保公共安全互操作性 V2（CSIv2）入站认证配置启用 CSIv2 身份声明。
 16. 可选：如果尚未安装和配置业务流程编排器资源管理器，那么现在可以对其进行配置。请执行第 188 页的『配置业务流程编排器资源管理器』。
 17. 可选：如果尚未安装和配置业务流程编排器观察器，那么现在可以对其进行配置。请执行第 191 页的『配置业务流程编排器观察器』。

结果

业务流程编排器已配置完毕。

bpeconfig.jacl 脚本文件

此脚本文件在服务器或集群中配置业务流程编排器以及所有必需的资源。

用途

此脚本能够以交互方式或批处理方式运行。它可以创建本地数据库以及必需的消息传递资源，并且可以选择配置业务流程编排器资源管理器和业务流程编排器观察器。

位置

bpeconfig.jacl 脚本文件在业务流程编排器的 config 目录中：

- 对于 Linux、UNIX 和 i5/OS 平台：在 *install_root/ProcessChoreographer/config* 目录中
- 对于 Windows 平台：在 *install_root\ProcessChoreographer\config* 目录中

限制

此脚本有下列限制：

对于 DB2 z/OS 版数据库

bpeconfig.jacl 脚本无法创建 DB2 z/OS 版数据库。您必须以手动方式创建数据库。

对于 DB2 数据库

如果选择 4 类通用驱动程序，那么 bpeconfig.jacl 脚本无法创建数据库，即使 DB2 安装在本地亦如此。

对于 Oracle 数据库

bpeconfig.jacl 脚本无法创建 Oracle 数据库。如果要让业务流程编排器使用 Oracle 数据库，那么必须以手动方式创建数据库。

对于 Microsoft SQL Server 数据库

bpeconfig.jacl 脚本无法创建远程数据库。要创建本地数据库，请使用 2 类 JDBC 驱动程序，并且不要指定 `-dbServerName` 参数。如果要让业务流程编排器使用远程 Microsoft SQL Server 数据库，那么必须以手动方式创建数据库。

在独立服务器环境中运行此脚本

在独立服务器环境中：

- 仅当应用程序服务器未处于运行状态时，才需要指定 `-conntype NONE` 选项。
- 如果服务器处于运行状态，并且已启用 WebSphere 管理安全性，那么请指定 `-user` 和 `-password` 选项。
- 如果未配置缺省概要文件，那么请添加 `-profileName` 选项。

在 Network Deployment 环境中运行此脚本

在 Network Deployment 环境中：

- 在 Deployment Manager 节点上运行此脚本。
- 仅当 Deployment Manager 未处于运行状态时，才需要指定 `-conntype NONE` 选项。
- 如果已启用 WebSphere 管理安全性，那么请指定 `-user` 和 `-password` 选项。
- 如果未配置缺省概要文件，那么请添加 `-profileName` 选项。

以非交互方式配置业务流程容器、业务流程编排器资源管理器和业务流程编排器观察器

如果在命令行上提供了必需的参数，那么不会提示您输入这些参数。要配置业务流程编排器，请输入下列其中一个命令：

在 Linux 和 UNIX 平台上，如果当前目录是 `install_root`，那么输入以下命令：

```
bin/wsadmin.sh -f ProcessChoreographer/config/bpeconfig.jacl parameters
```

在 i5/OS 平台上，如果当前目录是 `install_root`，那么输入以下命令：

```
bin/wsadmin -f ProcessChoreographer/config/bpeconfig.jacl parameters
```

在 Windows 平台上，如果当前目录是 `install_root`，那么输入以下命令：

```
bin\wsadmin -f ProcessChoreographer/config/bpeconfig.jacl parameters
```

其中，`parameters` 如下所示：

```

-conntype NONE
-user userName
-password userPassword
-profileName profileName
{-node nodeName -server serverName}
-cluster clusterName
-adminBFMUsers userList
-adminBFMGroups groupList
-monitorBFMUsers userList
-monitorBFMGroups groupList
-jmsBFMRunAsUser userID
-jmsBFMRunAsPwd password
-adminHTMUsers userList
-adminHTMGroups groupList
-monitorHTMUsers userList
-monitorHTMGroups groupList
-jmsHTMRunAsUser userID
-jmsHTMRunAsPwd password
-contextRootBFM contextRootBFM
-contextRootHTM contextRootHTM
-mailServerName mailServerName
-mailUser mailUserID
-mailPwd mailPassword
-hostName VirtualHostname
-explorerHost explorerURL
-remoteNode nodeName
-remoteServer serverName
-remoteCluster clusterName
-contextRootExplorer explorerContextRoot
-compileJSPs { yes | no }
-maxListEntries max
-createDB { yes | no }
-dbType databaseType
-dbVersion version
-dbHome databaseInstallPath
-dbJava JDBCDriverPath
-dbName databaseName
-dbUser databaseUser
-dbPwd databasePassword
-driverType JDBCDriverType
-dbTablespaceDir databaseTablespacePath
-dbServerName databaseServerName
-dbServerPort databaseServerPort
-dbStorageGroup DB2zOSSStorageGroup
-dbConnectionTarget DB2zOSSSubSystem
-dbSchema schemaQualifier
-dbInstance InformixInstance
-mqType JMSProviderType
-createQM { yes | no }
-qmNameGet getQueueManagerName
-mqClusterName mqClusterName
-qmNamePut putQueueManagerName
-mqHome MQInstallationDirectory
-mqUser JMSProviderUserID
-mqPwd JMSProviderPassword
-meStoreType { FILESTORE | DATASTORE }
-mqSchemaName mqSchemaName
-mqCreateTables { true | false }
-mqDataSource datasourceName
-medbUser meDatabaseUser
-medbPwd meDatabasePassword
-createEventCollector { yes | no }
-createObserver { yes | no }

```

注：以上某些参数是可选的，这取决于对其他参数指定的值。对于每个参数，以下内容描述了参数之间的依赖关系以及确定某个参数是可选还是必需的条件。对于命令行中未指定的任何必需参数，系统将按它们的列示顺序以交互方式提示您进行输入。

参数

使用 `wsadmin` 调用此脚本时，可以指定下列参数：

-conntype NONE

此参数指定没有管理连接可用。仅当应用程序服务器（对于独立应用程序服务器）或 Deployment Manager（对于 Network Deployment）未处于运行状态时，才应该指定此选项。

-user *userName*

如果已启用 WebSphere 管理安全性，那么必须提供用户标识以便进行认证。

-password *userPassword*

如果已启用 WebSphere 管理安全性，那么必须提供用户标识 *userName* 的密码。

-profileName *profileName*

其中 *profileName* 是用户定义的概要文件的名称。如果未配置缺省概要文件，那么应该指定此选项。概要文件必须已存在。

-node *nodeName*

其中 *nodeName* 是要配置业务流程编排器的节点的名称。如果只有一个节点，并且只有一个服务器，那么此参数是可选的。

-server *serverName*

其中 *serverName* 是要配置业务流程编排器的服务器的名称。如果只有一个节点，并且只有一个服务器，那么此参数是可选的。

-adminBFMUsers *userList*

其中 *userList* 是用户注册表中的用户名列表，Java 2 Enterprise Edition (J2EE) 角色 BPESystemAdministrator 将映射到这些用户。分隔符是竖线 (|)。要安装业务流程容器，此属性是必需的。此参数没有缺省值。必须设置 `adminBFMUsers` 和/或 `adminBFMGroups` 选项。

-adminBFMGroups *groupList*

其中 *groupList* 是用户注册表中的组名列表，J2EE 角色 BPESystemAdministrator 将映射到这些组。分隔符是竖线 (|)。要安装业务流程容器，此属性是必需的。此参数没有缺省值。必须设置 `adminBFMUsers` 和/或 `adminBFMGroups` 选项。

-monitorBFMUsers *userList*

其中 *userList* 是用户注册表中的用户名列表，J2EE 角色 BPESystemMonitor 将映射到这些用户。分隔符是竖线 (|)。要安装业务流程容器，此属性是必需的。此参数没有缺省值。必须设置 `monitorBFMUsers` 和/或 `monitorBFMGroups` 选项。

-monitorBFMGroups *groupList*

其中 *groupList* 是用户注册表中的组名列表，J2EE 角色 BPESystemMonitor 将映射到这些组。分隔符是竖线 (|)。要安装业务流程容器，此属性是必需的。此参数没有缺省值。必须设置 `monitorBFMUsers` 和/或 `monitorBFMGroups` 选项。

-jmsBFMRunAsUser *userID*

其中 *userID* 是 J2EE 角色 JMSAPIUser 在用户注册表中的 RunAs 用户标识。要配置业务流程容器，此属性是必需的。此参数没有缺省值。必须设置此参数。

-jmsBFMRunAsPwd *password*

其中 *password* 是 `jmsBFMRunAsUser` 用户标识的密码。要配置业务流程容器，此属性是必需的。此参数没有缺省值。必须设置此参数。

-adminHTMUsers *userList*

其中 *userList* 是用户注册表中的用户名列表，Java 2 Enterprise Edition (J2EE) 角色 `TaskSystemAdministrator` 将映射到这些用户。分隔符是竖线 (|)。要安装人员任务容器，此属性是必需的。此参数没有缺省值。必须设置 `adminHTMUsers` 和/或 `adminHTMGroups` 选项。

-adminHTMGroups *groupList*

其中 *groupList* 是用户注册表中的组名列表，J2EE 角色 `TaskSystemAdministrator` 将映射到这些组。分隔符是竖线 (|)。要安装人员任务容器，此属性是必需的。此参数没有缺省值。必须设置 `adminHTMUsers` 和/或 `adminHTMGroups` 选项。

-monitorHTMUsers *userList*

其中 *userList* 是用户注册表中的用户名列表，J2EE 角色 `TaskSystemMonitor` 将映射到这些用户。分隔符是竖线 (|)。要安装人员任务容器，此属性是必需的。此参数没有缺省值。必须设置 `monitorHTMUsers` 和/或 `monitorHTMGroups` 选项。

-monitorHTMGroups *groupList*

其中 *groupList* 是用户注册表中的组名列表，J2EE 角色 `TaskSystemMonitor` 将映射到这些组。分隔符是竖线 (|)。要安装人员任务容器，此属性是必需的。此参数没有缺省值。必须设置 `monitorHTMUsers` 和/或 `monitorHTMGroups` 选项。

-jmsHTMRunAsUser *userID*

其中 *userID* 是 J2EE 角色 `EscalationUser` 在用户注册表中的 `RunAs` 用户标识。要配置人员任务容器，此属性是必需的。此参数没有缺省值。必须设置此参数。

-jmsHTMRunAsPwd *password*

其中 *password* 是 `jmsHTMRunAsUser` 用户标识的密码。要配置人员任务容器，此属性是必需的。此参数没有缺省值。必须设置此参数。

-contextRootBFM *contextRootBFM*

其中 *contextRootBFM* 是 Web Service 端点 URL 的上下文根。对于业务流程管理器 (BFM) 而言，在服务器上，缺省上下文根是 `/BFMIF_${nodeName}_${serverName}`。在集群中，缺省上下文根是 `/BFMIF_clusterName`。

-contextRootHTM *contextRootHTM*

其中 *contextRootHTM* 是 Web Service 端点 URL 的上下文根。对于人员任务管理器 (HTM) 而言，在服务器上，缺省上下文根是 `/HTMIF_${nodeName}_${serverName}`。在集群中，缺省上下文根是 `/HTMIF_clusterName`。

-mailServerName *mailServerName*

其中 *mailServerName* 是人员任务管理器发送通知邮件时使用的邮件服务器主机名。配置邮件会话时，此参数是必需的。如果此参数设置为空值，那么将跳过邮件会话配置步骤。缺省值是本地主机的标准主机名。

-mailUser *mailUserID*

其中 *mailUserID* 是用于访问邮件服务器的用户标识。仅当邮件服务器要求进行认证时，才需要指定此参数。否则，可以省略此参数。要为人员任务管理器创建用于发送通知邮件的邮件会话，此参数是必需的。缺省值为空，仅当不需要进行认证时，此值才适用。

-mailPwd *mailPassword*

其中 *mailPassword* 是用户标识 *mailUserID* 的密码。仅当邮件服务器要求进行认证

时，才需要指定此参数。否则，可以省略此参数。要为人员任务管理器创建用于发送通知邮件的邮件会话，此参数是必需的。

-hostName *VirtualHostname*

其中，*VirtualHostname* 是将要运行业务流程编排器以及业务流程管理器 API 和人员任务管理器 API 的 Web Service 绑定的虚拟主机。缺省值是 `default_host`。

-maxListEntries *maximum*

其中，*maximum* 是业务流程编排器资源管理器对一个查询返回的最大结果数。缺省值是 10000。

-explorerHost *explorerURL*

其中 *explorerURL* 是业务流程编排器资源管理器的 URL。在非集群环境中，如果未指定此参数，那么将计算缺省值，例如 `http://localhost:9080`。人员任务管理器将使用此参数的值来链接到此资源管理器实例。

-precompileJSPs { no | yes }

确定是否对 Java Server Pages (JSP) 进行预编译。缺省值为 `no`。请注意，不可能对预编译的 JSP 进行调试。

-remoteNode *nodeName*

如果您不想连接至本地业务流程编排器配置，那么请使用此参数和 `remoteServer`。如果未指定此参数，那么缺省值为 `-node` 参数值。

-remoteServer *serverName*

如果您不想连接至本地业务流程编排器配置，那么请使用此参数和 `remoteNode`。如果未指定此参数，那么缺省值为 `-server` 参数值。

-remoteCluster *clusterName*

如果您不想连接至本地业务流程编排器配置，并且未指定 `remoteNode` 和 `remoteServer`，请使用此参数。如果未指定此参数，那么缺省值为 `-cluster` 参数值。

-contextRootExplorer *contextRootExplorer*

其中 *contextRootExplorer* 是业务流程编排器资源管理器的上下文根。缺省值是 `/bpc`，这将生成以下缺省 URL: `http://host:port/bpc`。上下文根在 WebSphere 单元中必须唯一。

-createDB { yes | no }

可能的值是 `yes` 或 `no`。如果设置为 `yes`，那么脚本将创建数据库。对于 z/OS 数据库和 Oracle 而言，此脚本无法创建数据库，而只能创建表空间和表。对于其他数据库类型而言，缺省值是 `yes`。对于生产系统，请使用 `no`。如果使用 `yes`，那么必须对从中调用 `bpeconfig.jacl` 的命令提示符设置适当的路径，以便运行相应的数据库命令，例如 `db2.exe`。

-dbType *databaseType*

其中 *databaseType* 是数据库类型。安装业务流程容器、创建数据库或数据库表以及创建数据源时，此参数是必需的。没有缺省值。可能的值是：

- Derby
- DB2
- zOS-DB2
- Informix
- iSeries-DB2
- MSSQL

- Oracle

-dbVersion *version*

其中 *version* 是数据库版本号。此参数没有缺省值。仅对于下列数据库类型，此参数才是必需的：

- 对于 DB2 z/OS 版而言，*version* 值必须是 7、8 或 9。
- 对于 Oracle 而言，*version* 值必须是 9 或 10。
- 对于 MSSQL 而言，如果数据库不支持 Unicode，那么 *version* 值必须是 2000；如果数据库支持 Unicode，那么必须是 2000U。

-dbHome *databaseInstallPath*

其中 *databaseInstallPath* 是数据库系统的安装目录。此参数仅对于 Informix 而言才是必需的；对于 DB2，如果 createDB 参数设置为 **Yes**，此参数是可选的。此参数用于创建数据库、数据库表和数据源。缺省值以及要求取决于数据库和平台：

对于 **DB2**:

- 在 Windows 平台上，缺省值是 *current_drive*\Program Files\IBM\SQLLIB，其中 *current_drive* 是当前盘符。
 - 在 Solaris 平台上，缺省值是 /export/home/\${dbUser}/sqllib。
 - 在其他平台上，缺省值是 /home/\${dbUser}/sqllib。
- 目录 \${dbHome}/bnd 和 \${dbHome}/bin 必须存在。

对于 **Informix**:

- 在 Windows 平台上，缺省值是 *current_drive*\Program Files\Informix，其中 *current_drive* 是当前盘符。
 - 在 Solaris 和 HP-UX 平台上，缺省值是 /opt/informix。
 - 在 Linux 和 AIX 平台上，缺省值是 /usr/informix。
- 文件 \${dbHome}/jdbc/lib/ifxjdbc.jar 必须存在。

-dbJava *JDBCDriverPath*

其中 *JDBCDriverPath* 是 JDBC 驱动程序所在的目录。仅对于数据库与驱动程序类型的下列组合，此参数才是必需的：

- DB2 通用数据库与 4 类驱动程序。缺省值是 *databaseInstallPath*/java。
- DB2 i5/OS 版与 2 类（本机）驱动程序。缺省值是 /QIBM/ProdData/Java400/ext。
- DB2 i5/OS 版与 4 类（Toolbox）驱动程序。缺省值是 /QIBM/ProdData/HTTP/Public/jt400/lib/java。
- DB2 z/OS 版与 4 类驱动程序。缺省值是 *databaseInstallPath*/java。
- Informix。缺省值是 *databaseInstallPath*/jdbc/lib。
- MSSQL DataSource 与 DataDirect 驱动程序类型。没有缺省值。
- Oracle。缺省值是 *databaseInstallPath*/jdbc/lib。

其中 *databaseInstallPath* 是数据库系统的安装目录。

在集群中以批处理方式运行此脚本时，如果数据库要求指定 -dbJava 参数，那么请按以下方式对每个主管集群成员的节点指定此参数：

`-dbJava.nodeName JDBCDriverPath_on_nodeName`

其中，*JDBCDriverPath* 是 JDBC 驱动程序的路径，*nodeName* 是节点的名称。

-dbName *databaseName*

其中 *databaseName* 是业务流程编排器数据库的名称。此参数用于创建数据库、数据库表和数据源。缺省值是 BPEDB。

- 对于 Oracle, 此参数的值为 TNS。
- 对于 Derby Network (并非嵌入式 Derby), 这必须是绝对路径名。
- 对于 i5/OS, 这是数据库名称或 IASP 硬件设备名。使用 Toolbox JDBC 驱动程序时, 缺省值为 *SYSBAS; 使用本机驱动程序时, 缺省值为 *LOCAL。

-dbUser *databaseUser*

其中 *databaseUser* 是用于访问数据库的用户标识。此参数用于创建数据源。缺省值取决于数据库和平台:

- 对于 Windows 平台上的 DB2: "db2admin"
- 对于其他平台上的 DB2: "db2inst1"
- 对于 Derby Network: 当前登录用户的用户标识
- 对于 Informix: "informix"
- 对于 Oracle: "system"
- 对于 MSSQL: 当前登录用户的用户标识

-dbPwd *databasePassword*

其中 *databasePassword* 是用户标识 *databaseUser* 的密码。

-driverType *JDBCDriverType*

其中 *JDBCDriverType* 是 JDBC 驱动程序的类型。此参数用于创建数据源。

- 对于 DB2, 可能的值是 Universal 或 CLI。
- 对于 i5/OS 上的 DB2: 可能的值是 native 或 toolbox。
- 对于 Derby, 可能的值是 Embedded 或 Network。
- 对于 Oracle, 可能的值是 oci8 或 thin。
- 对于 MSSQL, 可能的值是 Embedded 或 DataDirect。

-dbTablespaceDir *databaseTablespacePath*

其中 *databaseTablespacePath* 是用于创建数据库表空间的目录。此参数用于创建数据库和数据库表。仅对于下列数据库类型, 此参数才是必需的:

- 对于 Oracle, 没有缺省值。您必须提供值。
- 对于 DB2, 缺省值为空, 即不创建表空间。

-dbServerName *databaseServerName*

其中 *databaseServerName* 是业务流程编排器数据库所在的名称服务器。此参数用于创建数据源。

- 对于 DB2, 缺省值为空。对于 DB2 UDB, 此参数是可选的。如果未指定此参数, 那么将为 DB2 配置 2 类 JDBC 驱动程序, 否则将配置 4 类 JDBC 驱动程序。
- 对于 i5/OS 上的 DB2, 请指定服务器短名称。使用 Toolbox 驱动程序时, 缺省值是本地主机的短名称。
- 对于所有其他数据库类型, 缺省值是本地主机的标准主机名。

-dbServerPort *databaseServerPort*

其中 *databaseServerPort* 是业务流程编排器的数据库服务器的 TCP/IP 端口。如果指定了 dbServerName, 那么此参数是必需的。

- 对于 DB2, 缺省值是 50000。
- 对于 Derby Network, 缺省值是 1527。
- 对于 Informix, 缺省值是 1526。
- 对于 MSSQL, 缺省值是 1433。
- 对于驱动程序类型为 thin 的 Oracle, 缺省值是 1521。

-dbStorageGroup *DB2zOSSStorageGroup*

其中 *DB2zOSSStorageGroup* 是用于创建业务流程编排器数据库表的存储器组。仅对于 z/OS 平台上的 DB2, 此参数才是必需的。此参数没有缺省值, 并且不能为空。

-dbConnectionTarget *DB2zOSSubSystem*

其中 *DB2zOSSubSystem* 是用于创建业务流程编排器数据库表和数据源的 DB2 连接目标位置。仅对于 z/OS 平台上的 DB2, 此参数才是必需的。缺省值是 BPEDB。

-dbSchema *schemaQualifier*

对于 i5/OS, *schemaQualifier* 是集合名, 缺省值是 BPEDB。对于所有其他平台, *schemaQualifier* 是用于创建业务流程编排器数据库表和数据源的模式限定符。缺省值为空, 这表示使用依赖于所使用数据库类型的隐式模式限定符。

-dbInstance *InformixInstance*

其中 *InformixInstance* 是业务流程编排器 Informix 数据库的实例名。缺省值是 ids1。

-mqType *JMSProviderType*

其中 *JMSProviderType* 是用于业务流程编排器的 Java 消息服务 (JMS) 提供程序的类型。此参数用于创建队列管理器和队列、侦听器端口或激活规范以及队列连接工厂。

其中 *JMSProviderType* 是下列其中一个值:

WPM 表示缺省消息传递, 即 WebSphere 平台消息传递。此选项始终可用。

MQSeries

表示 WebSphere MQ。此选项要求安装 WebSphere MQ 产品。不推荐使用此值。

-createQM { *yes* | *no* }

指定此脚本是否创建本地 WebSphere MQ 队列管理器。仅当 *mqType* 参数值为 MQSeries 时, 此选项才有效 (不推荐使用此参数值)。此参数的缺省值是 *yes*。如果不想让此脚本创建 WebSphere MQ 队列管理器, 例如, 如果要在与运行此脚本的服务器不同的服务器上创建队列管理器, 那么使用 *no* 值。

-qmNameGet *getQueueManagerName*

其中 *getQueueManagerName* 是 GET 请求的队列管理器名。此参数用于创建队列管理器和队列以及侦听器端口和队列连接工厂。此参数不能包含 - 字符。*getQueueManagerName* 的缺省值是 *BPC_nodeName_serverName*。仅当 *mqType* 参数值为 MQSeries 时, 此选项才有效 (不推荐使用此参数值)。

-mqClusterName *mqClusterName*

其中 *mqClusterName* 是队列管理器将要加入的 WebSphere MQ 集群的名称。此参数是可选的。缺省值是 MQCluster。仅当 *mqType* 参数值为 MQSeries 时, 此选项才有效 (不推荐使用此参数值)。

-qmNamePut *putQueueManagerName*

其中 *putQueueManagerName* 是 PUT 请求的队列管理器名。仅当设置了

mqClusterName 参数时，才需要使用此参数。此参数用于创建队列管理器和队列以及侦听器端口和队列连接工厂。此参数不能包含 - 字符，并且不能与为 qmNameGet 参数指定的队列管理器名相同。putQueueManagerName 的缺省值是 BPCC_nodeName_serverName。

-mqHome *MQInstallationDirectory*

其中 *MQInstallationDirectory* 是 WebSphere MQ 的安装目录。此参数用于创建队列管理器和队列（仅限于 Windows 平台）以及侦听器端口和队列连接工厂。如果设置了 WebSphere 变量 MQ_INSTALL_ROOT，那么将使用该变量的值，并且不会对其进行修改。仅当 mqType 参数值为 MQSeries 时，此选项才有效（不推荐使用此参数值）。

如果未设置 MQ_INSTALL_ROOT，那么 *MQInstallationDirectory* 的缺省值视平台而定：

Windows 平台：

current_drive\Program Files\IBM\WebSphere MQ

AIX: /usr/mqm

i5/OS:

/QIBM/ProdData/mqm

Solaris、HP-UX 和 Linux:

/opt/mqm

-mqUser *JMSProviderUserID*

其中 *JMSProviderUserID* 是用于访问 JMS 提供程序的用户标识。

- 如果 mqType 值为 WPM，那么此参数用于对业务流程编排器 SI 总线执行认证；缺省值是当前登录用户。
- 在 Linux 和 UNIX 平台上，如果 mqType 值为 MQSeries，那么此参数用于创建队列管理器和队列。*JMSProviderUserID* 的缺省值是 mqm。

-mqPwd *JMSProviderPassword*

其中 *JMSProviderPassword* 是为 mqUser 提供的用户标识的密码。此参数没有缺省值。

-meStoreType { *FILESTORE* | *DATASTORE* }

设置业务流程编排器消息传递引擎的消息存储器类型。如果提供了 mqDataSource 参数，那么此参数设置为 DATASTORE。如果服务组件体系结构（SCA）正在使用 FILESTORE，那么此参数设置为 FILESTORE。在 Network Deployment 环境中，不支持 FILESTORE。如果未设置 mqDataSource，并且 SCA 使用 DATASTORE 作为它的消息存储器类型，那么将继承 SCA 消息传递引擎数据库设置，例如数据库类型、JDBC 提供程序和数据库服务器。在这种情况下，必须设置独立的数据库模式（请参阅下面的 mqSchemaName），并且还可以覆盖 mqCreateTables 标志。

-mqSchemaName *mqSchemaName*

其中 *mqSchemaName* 是缺省 JMS 提供程序消息传递引擎的数据库模式名。缺省值是 BPEME。仅当 meStoreType 设置为 DATASTORE 时，才使用此选项。

-mqCreateTables { *true* | *false* }

此布尔参数指定第一次连接时缺省 JMS 提供程序是否在消息传递引擎数据库中自动创建所需的表。如果对服务组件体系结构（SCA）将此标志设置为 true，那么此参

数也设置为 true。如果对 SCA 将此标志设置为 false，那么此参数也设置为 false。仅当 mqType 选项设置为 WPM 并且 meStoreType 设置为 DATASTORE 时，才使用此选项。

-medbUser *MEDBUserID*

其中 *MEDBUserID* 是用于访问消息传递引擎数据库的用户标识。此参数的缺省值是 dbUser 参数值。仅当 meStoreType 参数设置为 *DATASTORE*，并且未通过嵌入式 Derby JDBC 驱动程序来访问消息传递引擎数据库时，此参数才是必需的。

-medbPwd *MEDBPassword*

其中，*MEDBPassword* 是对 medbUser 参数提供的用户标识的密码。此参数没有缺省值。

-createEventCollector { *yes* | *no* }

以批处理方式运行时，缺省值是 yes，这将导致对业务流程编排器事件收集器应用程序进行配置，该应用程序是业务流程编排器观察器所必需的。使用此选项时，无法指定另一个数据库，缺省情况下将使用 BPEB 数据库，这意味着此选项不适用于高性能系统。如果您不想安装该应用程序，那么请将此参数值设置为 no。

-createObserver { *yes* | *no* }

以批处理方式运行时，缺省值是 yes，这将导致对业务流程编排器观察器应用程序进行配置。只能在非交互方式下使用此选项。使用此选项时，无法指定另一个数据库，缺省情况下将使用 BPEB 数据库，这意味着此选项不适用于高性能系统。如果您不想安装该应用程序，那么请将此参数值设置为 no。

以交互方式运行配置脚本

本示例演示通过运行 bpeconfig.jacl 脚本来安装和配置需要使用现有 DB2 数据库、人员任务容器和业务流程编排器资源管理器的业务流程容器。

限制：以交互方式运行时，此脚本无法配置业务流程编排器观察器和必需的事件收集器应用程序。如果要使用业务流程编排器观察器，那么必须执行第 191 页的『配置业务流程编排器观察器』。

1. 在服务器上（对于 Network Deployment，在 Deployment Manager 上），启动此脚本：

- 在 Linux 和 UNIX 平台上，输入以下命令：

```
install_root/bin/wsadmin.sh
    -f install_root/ProcessChoreographer/config/bpeconfig.jacl
    ( [-user userName] [-password password] | [-conntype NONE])
    [-profileName profileName]
```

- 在 i5/OS 平台上，输入以下命令：

```
install_root/bin/wsadmin
    -f install_root/ProcessChoreographer/config/bpeconfig.jacl
    ( [-user userName] [-password password] | [-conntype NONE])
    [-profileName profileName]
```

- 在 Windows 平台上，输入以下命令：

```
install_root\bin\wsadmin.bat
    -f install_root\ProcessChoreographer\config\bpeconfig.jacl
    ( [-user userName] [-password password] | [-conntype NONE])
    [-profileName profileName]
```

2. 以交互方式输入对所显示问题的回答：

- a. 在 Network Deployment 环境中, 将向您提供要在其中执行配置的服务器或集群。如果这不是正确的服务器或集群, 那么请输入 **No** 以获取下一个服务器或集群。如果这是正确的服务器或集群, 那么要求输入 **Yes**。
- b. 对于问题 `Install the business process container?`, 输入 **Yes**。
- c. 对于问题 `User(s) to add to role BPSystemAdministrator`, 输入具有业务流程管理员角色的用户的用户标识。
- d. 对于问题 `Group(s) to add to role BPSystemAdministrator`, 输入域用户注册表中映射到业务流程管理员角色的组。
- e. 对于问题 `User(s) to add to role BPSystemMonitor`, 输入具有业务流程监视员角色的用户的用户标识。
- f. 对于问题 `Group(s) to add to role BPSystemMonitor`, 输入域用户注册表中映射到业务流程监视员角色的组。
- g. 对于问题 `Run-as UserId for role JMSAPIUser`, 输入 `JMSAPIUser` 角色将使用的 `RunAs` 用户标识。
- h. 输入 `RunAs` 用户标识的密码。
- i. 对于问题 `Use WebSphere default messaging or WebSphere MQ [WPM/MQSeries]?`, 选择要使用的 `JMS` 提供程序。
- j. 输入以下内容:
 - 1) 对于问题 `Virtual Host for the SCA Web Service [default_host]:`, 请按 **Enter** 键以接受缺省值 `default_host` 作为服务组件体系结构 (SCA) Web 服务器虚拟主机。
 - 2) 对于问题 `Context root for the SCA Web Service [/BFMIF_PNODE_server1]:`, 请按 **Enter** 键以接受缺省值 `BFMIF_nodeName_serverName`。
- k. 对于问题 `Create the DataSource for the Process Choreographer database?`, 输入 **Yes**。
- l. 对于问题 `Create DataSource for a Derby, a DB2, an Informix, an Oracle, or an SQL Server database [Derby/DB2/zOS-DB2/iSeries-DB2/Informix/Oracle/MSSQL]?`, 对于本示例, 输入 **DB2**。如果选择其他数据库, 那么将显示其他特定于该数据库的问题。
- m. 输入数据库名称。
- n. 对于 `Database schema name (may be empty)` 提示, 请按 **Enter** 键以使用隐式模式限定符。
- o. 对于问题 `Universal or CLI?`, 请按 **Enter** 键以选择缺省的通用 `JDBC` 驱动程序。
- p. 对于问题 `DB2 User ID`, 输入用于访问数据库的用户标识。
- q. 输入该数据库用户标识的密码。
- r. 对于问题 `Database server name (may be empty, set to use the type 2 driver)`, 输入数据库所在服务器的名称。
- s. 对于问题 `Database server port`, 输入数据库服务器端口, 例如 `50000`。
- t. 对于 `JDBC driver directory on [yourHost]` 提示, 请输入 `DB2 JDBC` 驱动程序 `JAR` 文件所在的目录。
- u. 对于问题 `Create the Process Choreographer database objects?`, 如果当前登录用户标识有足够的权限来创建数据库, 并且已在当前环境中设置 `DB2` (例

如, “db2”可执行文件在 PATH 中), 那么可以输入 **Yes**。否则, 如果当前登录用户标识因没有足够的权限而无法创建数据库, 那么请输入 **No**。

如果回答 **Yes**:

- 1) 对于问题 DB2 tablespace directory (may be empty), 请按 **Enter** 键以保留为空。
- 2) 对于问题 Is 'BPEDB' an existing database (the Process Choreographer schema must not yet exist), 请根据您的环境进行回答。
- v. 如果显示了问题 User ID for access to Process Choreographer SI bus, 那么输入用于访问缺省 JMS 提供程序的用户标识。
- w. 输入 SI 总线认证用户标识的密码。
- x. 对于问题 Install the task container?, 输入 **Yes**。
- y. 对于问题 User(s) to add to role TaskSystemAdministrator, 输入具有任务管理员角色的用户的用户标识。
- z. 对于问题 Group(s) to add to role TaskSystemAdministrator, 输入域用户注册表中映射到任务管理员角色的组。
- aa. 对于问题 User(s) to add to role TaskSystemMonitor, 输入具有任务监视员角色的用户的用户标识。
- ab. 对于问题 Run-as UserID for role EscalationUser, 输入升级用户角色的 RunAs 用户标识, 例如 db2admin。
- ac. 输入升级用户标识的密码。如果使用第 157 页的 2g 步骤中的用户标识, 那么不会显示此提示。
- ad. 对于问题 Context root for the SCA Web Service [/HTMIF_nodeName_serverName]: , 输入服务组件体系结构 (SCA) Web 服务器的上下文根, 或者按 **Enter** 键以接受缺省值。
- ae. 对于问题 Create the mail notification session for the human task manager?, 如果您不想为人员任务管理器创建邮件通知会话, 那么输入 **No**。否则, 输入 **Yes** 并指定邮件传输主机。(可选) 可以指定用户标识和密码。
- af. 对于问题 Maximum number of list entries for the Process Choreographer Explorer, 按 **Enter** 键以使用缺省值 10000。
- ag. 对于问题 Context root for the Business Process Choreographer Explorer [/bpc]:, 输入业务流程编排器资源管理器的上下文根, 或者按 **Enter** 键以使用缺省值 /bpc。
- ah. 对于问题 Install the Business Process Choreographer Explorer?, 输入 **Yes** 以安装业务流程编排器资源管理器。然后, 对于问题 Precompile JSPs?, 如果要对 Java Server Pages (JSP) 进行预编译, 那么请输入 **Yes**, 否则输入 **No**。对于远程业务流程编排器资源管理器而言, 如果显示了问题 Node of Process Choreographer to connect to [PNODE]:, 那么输入要连接的业务流程编排器节点的名称; 如果显示了问题 Server of Process Choreographer to connect to [server1]:, 那么输入要连接的业务流程编排器服务器的名称, 或者按 **Enter** 键以接受缺省值。
- ai. 将显示各种信息, 例如业务流程编排器资源管理器的 URL。例如:

```
*****
* NOTE: The Process Choreographer URL will be used by the
* Human Task Manager on server server1 of node viennaNode01
```

```
* to link to this Explorer instance. Set an empty URL to not create this link.
*****
URL for this Process Choreographer Explorer [http://host_name:9080/bpc]:
```

输入此业务流程编排器资源管理器实例的 URL，或者按 **Enter** 以接受缺省值。

- aj. 将显示提示，指出可以用于配置业务流程编排器观察器的脚本文件所在的位置。

To interactively configure the EventCollector, please use the script
setupEventCollector located in *install_root*\ProcessChoreographer\config.
To interactively configure the Observer, please use the script
setupObserver located in *install_root*\ProcessChoreographer\config.

3. 如果发生问题，那么请检查日志文件。

日志文件

如果使用 bpeconfig.jacl 脚本文件创建配置时发生了问题，那么请检查下列日志文件：

- bpeconfig.log
- wsadmin.traceout

这两个文件都在概要文件的 logs 目录中：

- 对于 Linux、UNIX 和 iOS 平台：在 *profile_root*/logs 目录中
- 对于 Windows 平台：在 *profile_root*\logs 目录中

如果以连接方式运行此脚本，那么还需检查 SystemOut.log 和 SystemErr.log 文件。这两个文件在 logs 目录下与 wsadmin 脚本编制客户机所连接的应用程序服务器或 Deployment Manager 同名的子目录中。

相关任务

第 140 页的『使用 bpeconfig.jacl 脚本配置业务流程编排器』

本节描述如何使用 bpeconfig.jacl 脚本在给定的服务器或集群中配置业务流程编排器以及所有必需资源。

使用生成的 SQL 脚本为业务流程编排器创建数据库模式

bpeconfig.jacl 脚本将生成一个 SQL 脚本，此 SQL 脚本可用于为业务流程编排器创建数据库对象。

开始之前

您已使用 bpeconfig.jacl 脚本来配置业务流程编排器并使用了 -createDB no 选项来推迟创建数据库对象，或者 bpeconfig.jacl 脚本未能创建数据库。

关于此任务

您在配置业务流程编排器时提供的所有相关配置参数在生成的 SQL 文件中都已被替换。您可能需要用于高性能业务流程编排器配置的数据库，或者数据库管理员必须为您创建数据库，或者同时存在这两种情况。

过程

1. 在运行 bpeconfig.jacl 脚本的节点上查找所生成的 SQL 脚本。

选项	描述
对于 Linux 和 UNIX	<ul style="list-style-type: none"> • 如果已指定模式前缀，那么生成的脚本是： <i>profile_root/dbscripts/ProcessChoreographer/database_type/database_name/database_schema/createSchema.sql</i>。 • 如果未指定模式前缀，那么生成的脚本是： <i>profile_root/dbscripts/ProcessChoreographer/database_type/database_name/createSchema.sql</i>。
对于 i5/OS	生成的脚本是： <i>profile_root/dbscripts/ProcessChoreographer/database_type/collection_name/createSchema.sql</i> 。
对于 Windows	<ul style="list-style-type: none"> • 如果已指定模式前缀，那么生成的脚本是： <i>profile_root\dbscripts\ProcessChoreographer\database_type\database_name\database_schema\createSchema.sql</i>。 • 如果未指定模式前缀，那么生成的脚本是： <i>profile_root\dbscripts\ProcessChoreographer\database_type\database_name\createSchema.sql</i>。 <p>注：对于 SQL Server 而言，还有一个名为 <i>createSchemaUnicode.sql</i> 的版本，如果已对数据库进行 Unicode 配置，那么应该使用此版本。</p>
对于 z/OS	<p>提供了 ASCII SQL 脚本 <i>createSchema.sql</i> 以及等价的 EBCDIC DDL 脚本 <i>createSchema.ddl</i>：</p> <ul style="list-style-type: none"> • 如果已指定模式前缀，那么这两个文件都在 <i>profile_root/dbscripts/ProcessChoreographer/database_type/database_name/database_schema</i> 目录中。 • 如果未指定模式前缀，那么这两个文件都在 <i>profile_root/dbscripts/ProcessChoreographer/database_type/database_name</i> 目录中。

其中：

database_type

是下列其中一项：

- DB2
- DB2zOSV7
- DB2zOSV8
- Db2iSeries
- Derby

- Informix
- Oracle
- SQLServer

database_name
是数据库的名称。

database_schema
是模式的名称（如果正在使用模式的话）。

collection_name
是集合的名称；仅适用于 DB2 iSeries 版。

2. 如果该数据库尚未存在，并且不是 Derby 数据库，那么请数据库管理员根据第 95 页的『规划 BPEDB 数据库』和第 89 页的『规划安全性、用户标识和权限』中规划的值创建数据库和用户标识。
3. 对于远程数据库，请将生成的脚本复制到该远程节点。如果您无权执行此操作，那么请向数据库管理员提供一份脚本副本并与他讨论您的需求。
4. 您或数据库管理员必须对该 SQL 脚本进行定制：
 - a. 对于高性能系统，请指定第 95 页的『规划 BPEDB 数据库』的第 97 页的 5 步骤中规划的磁盘和表空间分配。
5. 使用下列其中一个命令在数据库主机上运行该 SQL 脚本：

选项	描述
对于 DB2 Linux 版、UNIX 版或 Windows 版	<code>db2 -tf createSchema.sql</code>
对于 DB2 iSeries 版	<code>db2 -tf createSchema.sql</code>
对于 DB2 z/OS 版	对于 ASCII 版本： <code>db2 -tf createSchema.sql</code> 对于 EBCDIC 版本： <code>db2 -tf createSchema.dll</code>
对于 Derby 数据库	<code>java -Dij.protocol=jdbc:derby: -Dij.database=BPEDB org.apache.derby.tools.ij createSchema.sql</code>
对于 Informix 数据库	<code>dbaccess databaseName createSchema.sql</code>
对于 Oracle 数据库	<code>sqlplus userID/password @database_name@createSchema.sql</code>
对于 SQL Server 数据库	对于 ASCII 数据库： <code>sqlcmd -U userID -P password -d database_name -i createSchema.sql</code> 对于 Unicode 数据库： <code>sqlcmd -U userID -P password -d database_name -i createSchemaUnicode.sql</code>

6. 配置 Java 数据库连接（JDBC）以访问远程数据库： 根据情况执行下列步骤：
 - 在集群中每个配置了业务流程编排器的成员上执行下列步骤。

- 在任何运行业务流程编排器并且没有本地数据库的服务器上执行下列步骤。
- 否则，如果服务器有本地数据库，那么不要执行下列步骤。
- a. 在应用程序服务器所在的服务器上安装合适的 2 类数据库客户机或 4 类 JDBC 驱动程序。
- b. 如果正在使用 2 类 JDBC 驱动程序，那么使数据库客户机知道新数据库。必须对该数据库进行编目，并使其可通过别名进行访问。如果正在使用 2 类 JDBC 驱动程序，那么请执行下列步骤，以使数据库客户机知道新数据库：

对于 Derby

由于只支持 4 类 JDBC 提供程序，因此此步骤不适用。

对于 DB2 通用数据库™

必须对该数据库进行编目，并且必须可通过别名访问该数据库。

对于 DB2 iSeries 版

必须对该数据库进行编目，并且必须可通过别名访问该数据库。

对于 DB2 z/OS 版

必须对该数据库进行编目，并且必须可通过别名访问该数据库。

对于 Informix Dynamic Server

由于只支持 4 类 JDBC 提供程序，因此此步骤不适用。

对于 Microsoft SQL Server

由于只支持 4 类 JDBC 提供程序，因此此步骤不适用。

对于 Oracle

将使用 TCP 网络服务名称 (TNS) 来访问数据库。

- c. 通过使用管理控制台，测试与该数据库的连接。

- 1) 单击**资源** → **JDBC** → **业务集成数据源**。
- 2) 必要时，选择另一个作用域并单击**应用**。

注：对于集群式业务流程编排器配置而言，请在集群级定义数据源。对于非集群式配置而言，应该在服务器级定义数据源。

- 3) 找到并选中 JNDI 名称为 jdbc/BPEDB 的数据源。
- 4) 单击**测试连接**。
- 5) 您应该会看到一条消息指出测试连接成功。

结果

业务流程编排器数据库已存在，并且您可以从任何配置了业务流程编排器的远程服务器或集群成员对其进行访问。

使用 SQL 脚本为业务流程编排器创建数据库

创建用于生产环境或高级拓扑的数据库，并使其可供远程访问。

关于此任务

系统提供了用于创建和管理所有受支持数据库系统的数据库模式的 SQL 脚本。通过使用此方法，可以让业务流程编排器使用自己的数据库，这对于提高性能而言至关重要。在独立服务器设置中，该数据库由一个应用程序服务器上的业务流程编排器配置专用。在集群式业务流程编排器设置中，该数据库由所有集群成员共享。

这些脚本为不同类型的数据库对象（例如模板数据和实例数据）分配独立的存储器。通过以此方式进行分隔，可以更好地进行负载均衡和性能调整。为了最大程度地提高性能，您可以在运行脚本前对其进行定制。

过程

1. 在数据库所在的服务器上，按照数据库系统的描述创建数据库。
 - 第 164 页的『为业务流程编排器创建 Derby 数据库』
 - 第 165 页的『为业务流程编排器创建 DB2 i5/OS 版数据库』
 - 第 165 页的『为业务流程编排器创建 DB2 Linux 版、UNIX 版和 Windows 版数据库』
 - 第 167 页的『为业务流程编排器创建 DB2 z/OS 版数据库』
 - 第 169 页的『为业务流程编排器创建 Informix Dynamic Server 数据库』
 - 第 170 页的『为业务流程编排器创建 Microsoft SQL Server 数据库』
 - 第 172 页的『为业务流程编排器创建 Oracle 数据库』
2. 配置 Java 数据库连接（JDBC）以访问远程数据库： 根据情况执行下列步骤：
 - 在集群中每个配置了业务流程编排器的成员上执行下列步骤。
 - 在任何运行业务流程编排器并且没有本地数据库的服务器上执行下列步骤。
 - 否则，如果服务器有本地数据库，那么不要执行下列步骤。
 - a. 在应用程序服务器所在的服务器上安装合适的 2 类数据库客户机或 4 类 JDBC 驱动程序。
 - b. 如果正在使用 2 类 JDBC 驱动程序，那么使数据库客户机知道新数据库。必须对该数据库进行编目，并使其可通过别名进行访问。如果正在使用 2 类 JDBC 驱动程序，那么请执行下列步骤，以使数据库客户机知道新数据库：
 - 对于 **Derby**
由于只支持 4 类 JDBC 提供程序，因此此步骤不适用。
 - 对于 **DB2 通用数据库**
必须对该数据库进行编目，并且必须可通过别名访问该数据库。
 - 对于 **DB2 iSeries 版**
必须对该数据库进行编目，并且必须可通过别名访问该数据库。
 - 对于 **DB2 z/OS 版**
必须对该数据库进行编目，并且必须可通过别名访问该数据库。
 - 对于 **Informix Dynamic Server**
由于只支持 4 类 JDBC 提供程序，因此此步骤不适用。
 - 对于 **Microsoft SQL Server**
由于只支持 4 类 JDBC 提供程序，因此此步骤不适用。
 - 对于 **Oracle**
将使用 TCP 网络服务名称（TNS）来访问数据库。
 - c. 通过使用管理控制台，测试与该数据库的连接。
 - 1) 单击资源 → **JDBC** → **业务集成数据源**。
 - 2) 必要时，选择另一个作用域并单击**应用**。

注：对于集群式业务流程编排器配置而言，请在集群级定义数据源。对于非集群式配置而言，应该在服务器级定义数据源。

 - 3) 找到并选中 JNDI 名称为 jdbc/BPEDB 的数据源。
 - 4) 单击**测试连接**。

5) 您应该会看到一条消息指出测试连接成功。

结果

业务流程编排器数据库已存在，并且您可以从任何配置了业务流程编排器的远程服务器或集群成员对其进行访问。

为业务流程编排器创建 Derby 数据库

通过本任务为业务流程编排器创建 Derby 数据库。

关于此任务

Derby 数据库系统随 WebSphere Process Server 一起交付。

要创建名为 BPEDB 的嵌入式 Derby 数据库，请执行下列操作：

过程

1. 通过执行下列其中一项操作，准备运行数据库创建脚本文件：

- 如果准备在缺省位置中创建数据库，那么以手动方式在适当的概要文件目录中创建 `databases` 子目录。在 Windows 平台上，请创建 `profile_root\databases`。在 Linux、UNIX 和 i5/OS 平台上，请创建 `profile_root/databases`。然后，切换到新目录。
- 如果要在非缺省位置中创建数据库，请切换到要创建新数据库的目录。

2. 将数据库创建脚本复制到步骤 1 中创建的目录。

- 在 Linux、UNIX 和 i5/OS 平台上，复制文件 `install_root/dbscripts/ProcessChoreographer/Derby/createDatabase.sql`。
- 在 Windows 平台上，复制文件 `install_root\dbscripts\ProcessChoreographer\Derby\createDatabase.sql`。

3. 检查在服务器上是否配置了 Java。输入命令：

```
java -version
```

如果显示了错误消息，那么在步骤 5 中运行数据库创建脚本时，必须在 Java 命令前面加上 Java 可执行文件的完整路径：

- 在 Windows 平台上，添加路径：`install_root\java\bin\`
- 在 Linux、UNIX 和 i5/OS 平台上，添加路径 `install_root/java/bin/`。

4. 按照数据库创建脚本 `createDatabase.sql` 开头的指示信息定制该脚本的副本。必须指定数据库的名称。在 Windows 平台上，请不要使用记事本编辑器，此编辑器无法以可读的格式显示脚本文件。

5. 创建数据库。在要创建数据库的目录中，按照数据库创建脚本文件 `createDatabase.sql` 开头描述的方式运行经过定制脚本文件版本。

结果

业务流程编排器的数据库已存在。

为业务流程编排器创建 DB2 i5/OS 版数据库

通过本任务为业务流程编排器创建 DB2 i5/OS 版数据库模式。

开始之前

已完成第 95 页的『规划 BPEDB 数据库』。

过程

1. 在主管数据库的机器上：如果不存在拥有该数据库的用户标识的集合，那么请创建一个集合。
2. 将模式创建脚本复制到主管数据库的机器。
 - 在 Linux、UNIX 和 i5/OS 平台上，此脚本是 `install_root/dbscripts/ProcessChoreographer/Db2iSeries/createSchema.sql`。
 - 在 Windows 平台上，此脚本是 `install_root\dbscripts\ProcessChoreographer\Db2iSeries\createSchema.sql`。
 - a. 如果要想业务流程编排器在第一次使用该数据库时创建缺省模式，并且您将在配置页面上选择**创建表**选项，那么请跳过此步骤。
 - b. 如果您打算**不选中创建表**选项，那么请创建 SQL 文件 `createSchema.sql` 的本地副本，然后根据第 95 页的『规划 BPEDB 数据库』和第 89 页的『规划安全性、用户标识和权限』中规划的值对其进行定制。
3. 创建数据库对象。在 Qshell 环境中，运行您的脚本副本。例如，如果该脚本在当前目录中，那么请输入以下命令：

```
db2 -tf createSchema.sql
```
4. 对于远程数据库，请使用 Toolbox JDBC 驱动程序。从数据库主机下载 JAR 文件 `/QIBM/ProdData/HTTP/Public/jt400/lib/jt400.jar`。
5. 对于本地数据库，请使用本机 JDBC 驱动程序。确保 classpath 包含 `/QIBM/ProdData/Java400/ext/db2_classes.jar`。

结果

业务流程编排器的 DB2 i5/OS 版模式已存在。

为业务流程编排器创建 DB2 Linux 版、UNIX 版和 Windows 版数据库

通过本任务为业务流程编排器创建 DB2 数据库。

开始之前

已完成第 95 页的『规划 BPEDB 数据库』。

过程

1. 在数据库所在的服务器上安装 DB2。
2. 在所有使用 2 类 Java 数据库连接 (JDBC) 驱动程序来访问数据库的远程应用程序服务器上安装 DB2 客户机。
3. 在应用程序服务器所在的服务器上，切换到所使用数据库系统的业务流程编排器配置脚本的目录：

- 在 Windows 系统上, 输入:


```
cd install_root\dbscripts\ProcessChoreographer\DB2
```
 - 在 UNIX 和 Linux 系统上, 输入:


```
cd install_root/dbscripts/ProcessChoreographer/DB2
```
4. 如果数据库与应用程序服务器在不同的服务器上:
 - a. 将下列 SQL 脚本复制到数据库服务器:


```
createDatabase.sql
createTablespace.sql
createSchema.sql
dropSchema.sql
dropTablespace.sql
```
 - b. 切换到已复制的 SQL 脚本所在的目录。
 5. 如果要使用现有数据库, 请跳转到 9 步骤以创建表空间和模式。Unicode 支持: 确保数据库支持 Unicode (UTF-8)。如果不支持 Unicode, 该数据库就无法存储所有在 Java 代码中处理的字符。并且, 如果客户机使用了不兼容的代码页, 那么可能会发生代码页转换问题。
 6. 在数据库服务器上创建 DB2 实例。
 7. 如果使用了对称多处理器 (SMP) 服务器, 那么请检查 DB2 可以使用的处理器数。检查许可证:
 - 在 AIX 系统上, 输入以下命令:


```
/usr/opt/db2_08_01/adm/db2licm -l
```
 - 在其他 UNIX 或 Linux 系统上, 输入以下命令:


```
/opt/IBM/db2/V8.1/adm/db2licm -l
```

必要时, 使用 db2clim 命令或 DB2 许可证中心来更改处理器许可证数。
 8. 创建新数据库:
 - a. 确保使用对数据库系统具有管理员权限的用户标识。
 - b. 如果要创建名为 BPEDB 的非生产数据库以便独立地进行开发、评估或演示, 请输入以下命令:


```
db2 -tf createDatabase.sql
```

对于生产环境, 请考虑使用专用的表空间容器并调整 DB2 参数。
 9. 要创建表空间和模式:
 - a. 根据第 95 页的『规划 BPEDB 数据库』中的规划, 定制 createTablespace.sql 表空间创建脚本的副本。
 - b. 确保您对数据库系统具有管理员权限。用于创建模式的用户标识必须是为业务流程容器配置数据源时指定的用户标识。
 - c. 确保已连接到正确的实例。请检查 DB2INSTANCE 环境变量。
 - d. 要连接到名为 *databaseName* 的数据库, 请在 DB2 命令行处理器中输入以下命令:


```
db2 connect to databaseName
```
 - e. 要创建表空间, 请输入以下命令:


```
db2 -tf createTablespace.sql
```

确保脚本输出未包含任何错误。如果发生了错误，那么可以使用 `dropTablespace.sql` 脚本删除该表空间。

- f. 要在 DB2 命令行处理器中创建模式（表、索引和视图），请输入以下命令：
- ```
db2 -tf createSchema.sql
```

确保脚本输出未包含任何错误。如果要删除该模式，请使用 `dropSchema.sql` 脚本。

**注：**如果现在不创建表空间和模式，那么以后必须使用**创建表**选项，以便业务流程编排器在第一次尝试使用该数据库时创建缺省表空间和模式。

10. 在每个以远程方式访问该数据库的应用程序服务器上：

- a. 通过输入以下命令，对数据库进行编目：

```
db2 catalog database databaseName as databaseAlias at node nodeName
```

有关进行数据库编目的更多信息，请参阅 DB2 文档。

- b. 通过输入下列命令，验证能否连接到该数据库：

```
db2 connect to databaseName user userID
db2 connect reset
```

## 结果

业务流程编排器的数据库已存在。

## 为业务流程编排器创建 DB2 z/OS 版数据库

通过本任务为业务流程编排器创建 DB2 z/OS 版数据库。

### 开始之前

已完成第 95 页的『规划 BPEDB 数据库』。

### 关于此任务

本主题描述如何创建 DB2 z/OS 版数据库以及如何验证能否从应用程序服务器所在的服务器访问该数据库。

### 过程

1. 必须已在 UNIX、Linux、Windows 或 i5/OS 服务器上安装 WebSphere Process Server。
2. 在数据库所在的 z/OS 服务器上：
  - a. 登录到本机 z/OS 环境。
  - b. 如果安装了多个 DB2 系统，那么请确定要使用的子系统。
  - c. 记录 DB2 子系统所侦听的 IP 端口。
  - d. 创建数据库和存储器组。请执行下列其中一项操作：
    - 使用 DB2 管理菜单来创建新的数据库和存储器组。
    - 按照 `createDatabase.sql` 脚本文件开头的指示信息编辑此文件的副本，然后运行该副本。此脚本文件在以下目录中：
      - 在 Windows 平台上：`install_root\dbscripts\ProcessChoreographer\database_type`

- 在 Linux、UNIX、i5/OS 平台上以及在 z/OS 平台上的 UNIX 系统服务 (USS) 中: `install_root/dbscripts/ProcessChoreographer/database_type`

其中, `database_type` 是下列其中一项:

- DB2zOSV7
- DB2zOSV8

记录名称。

- 确定连接到运行 WebSphere Process Server 的远程服务器上数据库时使用的用户标识。正常情况下, 为了确保安全, 此用户标识不是创建数据库时使用的用户标识。
- 授予该用户标识对该数据库和存储器组的访问权。该用户标识还必须有权为该数据库创建新表。
- 确定是否要在所连接用户标识的模式中创建表和视图以及是否要定制模式限定符。如果单个用户标识访问多个包含同名表的数据库, 那么必须使用不同的模式限定符, 以避免名称冲突。

3. 在 WebSphere Process Server 所在的服务器上:

- 确保已安装 DB2 Connect™ Gateway。DB2 Connect Gateway 包括在 DB2 UDB ESE 软件包中, 但也可以单独安装。

- 通过在 DB2 命令行窗口中使用下列命令, 对远程数据库进行编目:

```
catalog tcpip node zosnode remote host_name server IP_port ostype mvs
catalog database location as database_alias at node zosnode authentication dcs
catalog dcs database database_alias parms '.,,INTERRUPT_ENABLED'
```

- 确保指定了同步点管理器实例名。输入下列命令:

```
db2 update dbm cfg using SPM_NAME host_name
db2 update dbm cfg using SPM_LOG_FILE_SZ log_file_size
```

- 注意下列信息:

在 DB2 z/OS 版与 DB2 Linux 版、UNIX 版和 Windows 版之间存在着重要差别。DB2 Linux 版、UNIX 版和 Windows 版中没有子系统的概念, 但 DB2 z/OS 版中有此概念。为了避免将数据库名称与子系统名称混淆, 了解下面这一点十分重要: 由于 DB2 z/OS 版在子系统中运行, 因此 `catalog node` 和 `catalog database` 命令必须标识相应的子系统。在 DB2 Linux 版、UNIX 版和 Windows 版中, 由于子系统名称不是已知的概念, 因此 `catalog` 命令所链接的数据库名称实际上是 DB2 z/OS 版子系统的名称。

- 通过输入以下命令, 验证能否与远程子系统建立连接:

```
db2 connect to subsystem user userid using password
```

- 在应用程序服务器所在的服务器上, 切换到所使用数据库系统的业务流程编排器配置脚本的目录:

- 在 Windows 系统上, 根据 DB2 版本的不同, 输入下列其中一个命令:

```
cd install_root\dbscripts\ProcessChoreographer\DB2zOSV7
cd install_root\dbscripts\ProcessChoreographer\DB2zOSV8
```

- 在 UNIX、Linux 和 i5/OS 系统上, 根据 DB2 版本的不同, 输入下列其中一个命令:

```
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV7
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV8
```



- g. 根据第 95 页的『规划 BPEDB 数据库』中的规划，定制 createTablespace.sql 表空间创建脚本的副本。将 @STOGRP@ 替换为存储器组名，并将 @DBNAME@ 替换为数据库名称而非子系统名称。
- h. 对 createTablespace.sql 脚本进行定制后，按照此脚本开头的指示信息运行此脚本。如果要删除表空间，请定制并运行 dropTablespace.sql 脚本。
- i. 编辑 createSchema.sql 脚本。
  - 1) 将 @STOGRP@ 替换为存储器组名。
  - 2) 将 @DBNAME@ 替换为数据库名称而非子系统名称。
  - 3) 将 @SCHEMA@ 替换为模式限定符，或者从脚本中除去 @SCHEMA@ 以及跟在后面的点。定制模式限定符只能与 DB2 通用 JDBC 驱动程序配合使用，并要求将 customSQLID 配置属性设置为适当的值。
- j. 对 createSchema.sql 脚本进行定制后，按照此脚本开头的指示信息运行此脚本。如果此脚本无法正常运行，或者您想除去表和视图，那么请使用 dropSchema.sql 脚本删除该模式。但是，运行此脚本前，请替换 @SCHEMA@。

## 结果

业务流程编排器的数据库已存在。

## 为业务流程编排器创建 Informix Dynamic Server 数据库

通过本任务为业务流程编排器创建 Informix Dynamic Server 数据库。

### 开始之前

已完成第 95 页的『规划 BPEDB 数据库』。

### 过程

1. 在数据库所在的计算机上安装 Informix 服务器。
2. 创建 Informix 服务器实例。确保正确地设置 Informix 环境变量。尤其是，*INFORMIXSERVER* 必须指向新实例，并且 *ONCONFIG* 必须指向该实例的配置文件。有关不同环境变量以及如何进行设置的更多详细信息，请参阅 Informix Dynamic Server 的文档。请确保将全球语言支持（GLS）的相关环境变量设置为支持 Unicode（UTF-8）。要存储所有可以在 Java 代码中处理的字符，必须支持 Unicode。
3. 在下列服务器上复制并配置 Java 数据库连接（JDBC）驱动程序：
  - 所有使用该数据库服务器的远程应用程序服务器。
4. 在应用程序服务器所在的服务器上，切换到所使用数据库系统的业务流程编排器配置脚本的目录：
  - 在 Windows 系统上，输入：

```
cd install_root\dbscripts\ProcessChoreographer\Informix
```
  - 在 UNIX 和 Linux 系统上，输入：

```
cd install_root/dbscripts/ProcessChoreographer/Informix
```
5. 如果数据库与应用程序服务器在不同的服务器上：
  - a. 将操作系统的脚本从应用程序服务器上的 ProcessChoreographer 目录复制到数据库所在服务器上的适当目录：对于 Linux 和 UNIX 上的 Informix，复制下列文件：



```
createDatabase.sql
createSchema.sql
dropSchema.sql
createDbSpace.sh
dropDbSpace.sh
```

对于 Windows 上的 Informix, 复制下列文件:

```
createDatabase.sql
createSchema.sql
dropSchema.sql
createDbSpace.bat
dropDbSpace.bat
```

- b. 切换到已复制的 SQL 脚本所在的目录。
6. 如果要使用缺省设置来创建适合于独立地进行开发、评估或演示工作的非生产数据库, 那么输入以下命令:

```
dbaccess - createDatabase.sql
```

此命令将为所使用的用户标识创建 Informix 数据库 BPEDB。确保脚本输出未包含任何错误。可以使用 dropSchema.sql 脚本仅删除模式, 也可以使用 SQL 命令 DROP DATABASE 删除整个数据库。

7. 如果要为生产系统创建数据库, 那么必须以手动方式创建该数据库:
  - a. 创建一个数据库, 例如将其命名为 BPEDB。
  - b. 创建数据库的 Dbspaces。

在 Windows 系统上, 阅读 createDbSpace.bat 文件中的指示信息。将此脚本中的参数值调整为适合于环境的值。

在 UNIX 和 Linux 系统上, 阅读 createDbSpace.sh 文件中的指示信息。将此脚本中的参数值调整为适合于环境的值。

- c. 通过输入以下命令运行此脚本, 以创建模式:

```
dbaccess databaseName createSchema.sql
```

其中 *databaseName* 是数据库名称, 例如 BPEDB。

- d. 检查脚本输出是否包含任何错误。如果要删除该模式, 请使用 dropSchema.sql 脚本。

## 结果

业务流程编排器的数据库已存在。

## 为业务流程编排器创建 Microsoft SQL Server 数据库

通过本任务为业务流程编排器创建 Microsoft SQL Server 数据库。

### 开始之前

已完成第 95 页的『规划 BPEDB 数据库』。

### 过程

1. 在数据库所在的服务器上安装 Microsoft SQL Server。确保选择了创建区分大小写实例的选项。如果已有在创建时指定了不区分大小写选项的 SQL Server, 那么必须运行重建主工具并将整理设置更改为区分大小写。确保该服务器支持 Unicode。

2. 确保数据库服务器和分布式事务协调程序 (DTC) 处于运行状态。确保已针对分布式事务配置数据库服务器。有关详细信息, 请参阅 Microsoft SQL Server 的文档。
3. 在应用程序服务器所在的服务器上, 切换到所使用数据库系统的业务流程编排器配置脚本的目录:
  - 在 Windows 系统上, 输入:

```
cd install_root\dbscripts\ProcessChoreographer\SQLServer
```
  - 在 UNIX 和 Linux 系统上, 输入:

```
cd install_root/dbscripts/ProcessChoreographer/SQLServer
```
4. 如果数据库与应用程序服务器在不同的服务器上:
  - a. 将数据库的 SQL 脚本从应用程序服务器上的 SQLServer 目录复制到数据库所在服务器上的适当目录:

```
createDatabase.sql
createDatabaseUnicode.sql
createSchema.sql
createSchemaUnicode.sql
dropSchema.sql
```
  - b. 切换到已复制的 SQL 脚本所在的目录。
5. 请执行下列其中一项操作:
  - 如果要创建名为 『BPEDB』 的非生产 SQL Server 数据库以便独立地进行开发、评估或演示:
    - a. 运行下列其中一个脚本并按照脚本开头的指示信息执行操作。
      - createDatabase.sql
      - createDatabaseUnicode.sql (用于 Unicode 数据库)例如, 输入以下命令:

```
sqlcmd -U userID -P password -i createDatabase.sql
```
  - b. 确保脚本输出未包含任何错误。如果发生了错误, 那么可以使用 dropSchema.sql 脚本来删除该模式。
- 如果要创建 SQL Server 生产数据库, 那么以手动方式创建该数据库:
  - a. 创建一个数据库, 例如将其命名为 『BPEDB』。
  - b. 要创建模式, 请运行下列其中一个脚本并按照脚本开头的指示信息执行操作。
    - createSchema.sql
    - createSchemaUnicode.sql (如果已创建 Unicode 数据库)例如, 输入以下命令:

```
sqlcmd -U userID -P password -i createSchema.sql
```
- c. 确保脚本输出未包含任何错误。如果发生了错误, 那么可以使用 dropSchema.sql 脚本来删除该模式。

## 结果

业务流程编排器的数据库已存在。

## 为业务流程编排器创建 Oracle 数据库

通过本任务为业务流程编排器创建 Oracle 数据库。

### 开始之前

已完成第 95 页的『规划 BPEDB 数据库』。

### 过程

1. 在数据库所在的计算机上安装 Oracle 服务器。请确保使用 lib32 子目录中的 32 位 Oracle 库。
2. 在 Linux 和 UNIX 系统上，确保为 root 用户设置了 `ORACLE_BASE` 和 `ORACLE_HOME` 环境变量。
3. 检查类路径，确保 JDBC 驱动程序正在使用 `ojdbc14.jar` 文件。
4. 在 Linux 和 UNIX 系统上，创建软链接以指向 `/usr/lib` 目录中的下列 Oracle 库：
  - 对于 Oracle 10g: 链接到 `libclnt.so.10.1`。
  - 对于 Oracle 9i: 链接到 `libnnz10.so`、`libclnt.so.10.1`、`libclntsh.so.10.1` 和 `libocijdbc10.so`。

有关如何设置 Oracle OCI 客户机的更多详细信息，请参阅 Oracle 提供的文档。

5. 使用数据库配置助手创建一个 Oracle 数据库，例如将其命名为 BPEDB。产品未提供为业务流程编排器快速创建缺省 Oracle 数据库的脚本。请确保对此数据库选择 JServer 选项。创建的此数据库必须使用 Unicode 代码页。
6. 通过输入以下命令，启动 Oracle 侦听器：

```
lsnrctl start
```
7. 可选：如果您不想定制表空间和模式，那么可以跳过本任务中的其余步骤，在这种情况下，业务流程编排器在第一次尝试使用该数据库时将创建缺省表空间和模式。
8. 在应用程序服务器所在的服务器上，切换到所使用数据库系统的业务流程编排器配置脚本的目录：

- 在 Windows 系统上，输入：

```
cd install_root\dbscripts\ProcessChoreographer\Oracle
```

- 在 UNIX 和 Linux 系统上，输入：

```
cd install_root/dbscripts/ProcessChoreographer/Oracle
```

9. 如果数据库与应用程序服务器在不同的服务器上：
  - a. 将下列 Oracle 配置 SQL 脚本从应用程序服务器所在服务器上的业务流程编排器子目录复制到数据库所在服务器上的相应目录：

```
createSchema.sql
createTablespace.sql
dropSchema.sql
dropTablespace.sql
```

- b. 在数据库服务器上，切换到已复制的 SQL 脚本所在的目录。

10. 如果您不希望在缺省实例中创建模式，请将 `ORACLE_SID` 环境变量设置为步骤 5 中创建的数据库的 SID。
11. 确保运行这些脚本的用户至少具有下列数据库特权：CREATE SESSION、CREATE TABLESPACE、DROP TABLESPACE、CREATE TABLE 和 CREATE VIEW。

12. 根据第 95 页的『规划 BPEDB 数据库』中的规划，定制 createTablespace.sql 表空间创建脚本的副本。

13. 要创建表空间，请运行 createTablespace.sql 脚本。如果要进行测试，可以对所有表空间使用同一位置并将路径作为命令行参数传递给该脚本。例如，在 Windows 系统上，如果使用用户标识“bpeuser”、密码“bpepwd”、数据库名称“BPEDB”和表空间路径 d:\mydb\ts，那么输入以下命令：

```
sqlplus bpeuser/bpepwd@BPEDB @createTablespace.sql d:\mydb\ts
```

如果要删除表空间，可以使用 dropTablespace.sql 脚本。

14. 确保将拥有这些表的用户在上一步骤中创建的所有表空间中被授予足够的限额。

15. 按照模式创建脚本 createSchema.sql 开头的指示信息编辑此文件，并将占位符 @SCHEMA@ 替换为模式名。如果 @SCHEMA@ 与运行 createSchema.sql 脚本的用户不同，那么确保此用户具有下列数据库特权：CREATE ANY TABLE、ALTER ANY TABLE、CREATE ANY INDEX 和 CREATE ANY VIEW。

16. 要创建模式，请运行 createSchema.sql 脚本。例如，在 Windows 系统上，输入：

```
sqlplus bpeuser/bpepwd@BPEDB @createSchema.sql
```

## 结果

业务流程编排器的数据库已存在。

---

## 配置人员目录提供程序

通过本任务来配置轻量级目录访问协议（LDAP）或虚拟成员管理器（VMM）人员目录提供程序，业务流程编排器将使用此提供程序来确定能够启动流程或者声明活动/任务的人员。

### 关于此任务

每种受支持的人员目录服务都需要相应的人员目录提供程序。支持下列人员目录提供程序：

表 14. 支持的人员目录提供程序

| 人员目录提供程序                           | 人员目录提供程序选项    |
|------------------------------------|---------------|
| 轻量级目录访问协议（LDAP）目录                  | LDAP 人员目录提供程序 |
| 虚拟成员管理器                            | VMM 人员目录提供程序  |
| 本地操作系统用户注册表                        | 系统人员目录提供程序    |
| WebSphere Application Server 用户注册表 | 用户注册表人员目录提供程序 |

所有这些插件都已随缺省配置一起安装。可以将用户注册表和系统插件与缺省配置配合使用。对于 VMM 而言，缺省配置通常已足够。

## 配置虚拟成员管理器人员目录提供程序

通过本任务来配置虚拟成员管理器（VMM）人员目录提供程序，业务流程编排器将使用此提供程序来执行人员分配，从而确定能够启动流程或者声明活动/任务的人员。缺省人员目录提供程序已可供使用，仅当您引入了定制人员分配条件时才需要对其进行配置。

## 开始之前

已配置联合存储库。

## 过程

1. 备份 VMM 的标准变换文件并对其指定另一个名称，例如 myVMMTransformation.xml。
  - 在 Windows 平台上，此文件是 `install_root\ProcessChoreographer\Staff\VMMTransformation.xml`
  - 在 Linux、UNIX 和 i5/OS 平台上，此文件是 `install_root/ProcessChoreographer/Staff/VMMTransformation.xml`

**警告：** 请不要修改变换文件的原始版本，因为如果这样做，将来应用服务包或修订包时将覆盖该版本，而不会发出警告。

2. 如果业务流程编排器被配置在集群中，那么请将该变换文件的副本存放所安装的每个主管集群成员的 WebSphere Process Server 上。
3. 在管理控制台中，单击资源 → 人员目录提供程序。
4. 选择适当的节点。

| 选项                                               | 描述                                                                           |
|--------------------------------------------------|------------------------------------------------------------------------------|
| 对于独立概要文件:                                        | 只显示了一个节点。                                                                    |
| 对于在单服务器上配置业务流程编排器的 <b>Network Deployment</b> 环境: | 选择包含该服务器的节点。                                                                 |
| 对于在集群中配置业务流程编排器的 <b>Network Deployment</b> 环境:   | 必须在每个主管集群成员的节点上配置人员目录提供程序。请选择第一个节点，在该节点上配置人员目录提供程序，然后对所有其他主管集群成员的节点重复执行配置工作。 |

5. 要创建新的 VMM 人员目录配置，请执行下列操作：
  - a. 单击 **VMM 人员目录提供程序**。
  - b. 在**其他属性**中，选择**人员目录配置**。
  - c. 单击**新建** → **浏览**，然后选择您在 1 步骤中复制的可扩展样式表语言（XSL）变换文件副本。
  - d. 单击**下一步**。
  - e. 在**常规属性**部分中，输入新人员目录配置的管理名。
  - f. 可选： 输入描述。
  - g. 输入用于向系统标识此配置的唯一 Java 命名和目录接口（JNDI）名称。例如，`bpe/staff/myvmmconfiguration`。

**注：** 没有其他配置参数。

  - h. 单击**确定**，然后单击**保存**。
6. 要激活提供程序配置，请停止配置了提供程序的服务器，然后再将它们启动。
7. 可选： 如果要更改 XSL 变换文件，请执行下列操作：
  - a. 根据需要，编辑该 XSL 文件。
  - b. 重新启动服务器。在 **Network Deployment** 环境中，必须使新版本的 XSL 文件可供所有服务器使用，并且必须重新启动所有服务器。

8. 如果任何步骤发生问题，请参阅 PDF 文档 *Troubleshooting WebSphere Process Server*。

## 结果

VMM 人员目录提供程序已配置完毕。

## 配置 LDAP 人员目录提供程序

通过本任务来配置轻量级目录访问协议 (LDAP) 人员目录提供程序，业务流程编排器将使用此提供程序来执行人员分配，从而确定能够启动流程或者声明活动/任务的人员。

### 开始之前

已执行第 106 页的『规划人员目录提供程序』描述的 LDAP 规划工作。

### 关于此任务

LDAP 人员目录提供程序配置在初始化完毕后，其 URL 指向本地 LDAP 服务器。随后，必须将此 URL 更改为指向实际的 LDAP 服务器，该服务器通常不在应用程序服务器的本地位置。LDAP 人员目录提供程序针对允许进行匿名访问的 LDAP 服务器进行配置。

### 过程

1. 备份 LDAP 的标准变换文件并对其指定另一个名称，例如 myLDAPTransformation.xml。
  - 在 Windows 平台上，此文件是 `install_root\ProcessChoreographer\Staff\LDAPTransformation.xml`
  - 在 Linux、UNIX 和 i5/OS 平台上，此文件是 `install_root/ProcessChoreographer/Staff/LDAPTransformation.xml`
2. 可选：修改您的变换文件副本，使其与贵公司的 LDAP 模式相符。现在，您可以第 176 页的『修改 LDAP 变换文件』，也可以推迟执行此操作。  
**警告：** 请不要修改变换文件的原始版本，因为如果这样做，将来应用服务包或修订包时将覆盖该版本，而不会发出警告。
3. 如果业务流程编排器被配置在集群中，那么请将该变换文件的副本存放到所安装的每个主管集群成员的 WebSphere Process Server 上。
4. 在管理控制台中，单击资源 → 人员目录提供程序。
5. 选择适当的节点。

| 选项                                               | 描述                                                                           |
|--------------------------------------------------|------------------------------------------------------------------------------|
| 对于独立概要文件:                                        | 只显示了一个节点。                                                                    |
| 对于在单服务器上配置业务流程编排器的 <b>Network Deployment</b> 环境: | 选择包含该服务器的节点。                                                                 |
| 对于在集群中配置业务流程编排器的 <b>Network Deployment</b> 环境:   | 必须在每个主管集群成员的节点上配置人员目录提供程序。请选择第一个节点，在该节点上配置人员目录提供程序，然后对所有其他主管集群成员的节点重复执行配置工作。 |



6. 要在所选节点上创建新的 LDAP 配置:
  - a. 单击 **LDAP 人员目录提供程序**。
  - b. 在“其他属性”中, 单击**人员目录配置**。
  - c. 单击**新建** → **浏览**, 然后选择变换文件副本。
  - d. 单击**下一步**。
  - e. 输入人员目录配置的管理名。
  - f. 输入描述。
  - g. 输入人员任务用于引用此提供程序的 Java 命名和目录接口 (JNDI) 名称。例如, bpe/staff/ldapservlet1。
  - h. 单击**应用**。
  - i. 单击**定制属性**。
  - j. 对于每个必需属性以及第 106 页的 2 中规划的任何可选属性, 请单击该属性的名称, 输入值, 然后单击**确定**。

**注:** 对于其他可选属性, 可以设置对 JNDI 定义的属性, 例如启用 LDAP 推荐。对于 providerURL, 可以指定以 ldap:// 或 ldaps:// 开头的 URL。
  - k. 要应用更改, 请单击**保存**。
7. 可选: 如果要更改 XSL 变换文件, 请执行下列操作:
  - a. 根据需要, 编辑该 XSL 文件。
  - b. 重新启动服务器。在 Network Deployment 环境中, 必须使新版本的 XSL 文件可供所有服务器使用, 并且必须重新启动所有服务器。
8. 要激活提供程序配置, 请停止配置了提供程序的服务器, 然后再将它们启动。
9. 如果任何步骤发生问题, 请参阅 PDF 文档 *Troubleshooting WebSphere Process Server*。

## 结果

现在, 人员任务和流程可以使用人员分配服务来解析人员分配查询以及确定各位人员可以执行的活动。

## 修改 LDAP 变换文件

本节描述如何修改 LDAP 变换 XSL 文件, 以使其适合于贵公司的 LDAP 模式。

缺省的 LDAPTransformation.xml 文件将预定义人员分配条件映射到 LDAP 查询, 后者使用 WebSphere 所采用的缺省 LDAP 模式的元素。此模式作了如下假定:

- 组条目的 LDAP 对象类是 groupOfName。
- 包含组成员 DN 的组条目属性是 member。
- 人员条目的 LDAP 对象类是 inetOrgPerson。
- 人员条目中包含登录标识的属性是 uid。
- 包含人员电子邮件地址的人员条目属性是 mail。
- 包含人员的经理专有名称的人员条目属性是 manager。

如果您的 LDAP 模式使用其他对象类和属性名, 那么必须在所使用的 LDAP 变换文件中更改这些设置。请备份原始 LDAPTransformation.xml 文件, 如第 175 页的『配置 LDAP 人员目录提供程序』所述。



**警告:** 请不要修改变换文件的原始版本, 因为如果这样做, 将来应用服务包或修订包时将覆盖该版本, 而不会发出警告。

通常, 通过编辑该文件的变量声明部分更改所有人员分配条件的设置已足够:

```
<xsl:variable name="DefaultGroupClass">groupOfNames</xsl:variable>
<xsl:variable name="DefaultGroupClassMemberAttribute">member</xsl:variable>

<xsl:variable name="DefaultPersonClass">inetOrgPerson</xsl:variable>
<xsl:variable name="DefaultUserIDAttribute">uid</xsl:variable>
<xsl:variable name="DefaultMailAttribute">mail</xsl:variable>
<xsl:variable name="DefaultManagerAttribute">manager</xsl:variable>
```

可以在用于转换各个人员分配条件的 XSL 模板中应用更改, 如以下示例所示。

### 示例: GroupMembers

将组条目的对象类更改为 `groupOfUniqueNames`, 将包含成员 DN 列表的组条目属性更改为 `uniqueMember`, 并将包含登录标识的人员条目属性更改为 `cn`:

```
<ldap:usersOfGroup>
...

<ldap:attribute>
<xsl:attribute name="name">uniqueMember</xsl:attribute>
<xsl:attribute name="objectclass">groupOfUniqueNames</xsl:attribute>
<xsl:attribute name="usage">recursive</xsl:attribute>
</ldap:attribute>

...
<ldap:attribute>
<xsl:attribute name="name">cn</xsl:attribute>
<xsl:attribute name="objectclass">inetOrgPerson</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>
</ldap:attribute>
...
<ldap:resultObject>
<xsl:attribute name="objectclass">groupOfUniqueNames</xsl:attribute>
<xsl:attribute name="usage">recursive</xsl:attribute>
<ldap:resultAttribute>
<xsl:attribute name="name">uniqueMember</xsl:attribute>
<xsl:attribute name="destination">intermediate</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>

<ldap:resultObject>
<xsl:attribute name="objectclass"><xsl:value-of select="$DefaultPersonClass"/></xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">cn</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>

</ldap:usersOfGroup>
```

### 示例: GroupMembersWithoutFilteredUsers

将 LDAP 过滤器运算符更改为 `>=`。

```
<ldap:StaffQueries>
<ldap:usersOfGroup>
...
</ldap:usersOfGroup>
```

```

<ldap:intermediateResult>
<xsl:attribute name="name">filteredusers</xsl:attribute>
<ldap:search>
<xsl:attribute name="filter">
<xsl:value-of select="staff:parameter[@id='FilterAttribute']"/>
=>
<xsl:value-of select="staff:parameter[@id='FilterValue']"/>
</xsl:attribute>
...
<ldap:search>
...
</ldap:intermediateResult>
...
</ldap:StaffQueries>

```

## 示例: GroupSearch

将搜索属性更改为 MyType, 将对象类更改为 mypersonclass, 将包含登录标识的属性更改为 myuid。

```

<ldap:StaffQueries>
...
<ldap:search>
<xsl:attribute name="filter">
(&
...
<xsl:if test="staff:parameter[@id='MyType']!="">
(<xsl:value-of select="$GS_Type"/>=
<xsl:value-of select="staff:parameter[@id='Type']"/>)
)
...
</xsl:attribute>

<ldap:attribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>
</ldap:attribute>
...
<ldap:resultObject>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>

<ldap:search>
</ldap:StaffQueries>

```

## 示例: Manager of Employee

将包含经理 DN 的属性更改为 managerentry, 并将经理登录标识属性的源更改为 name。

```

<ldap:StaffQueries>

<ldap:intermediateResult>
...
<ldap:user>
...
<xsl:attribute name="name">managerentry</xsl:attribute>
...
<ldap:resultObject>
<xsl:attribute name="objectclass"><xsl:value-of select="$DefaultPersonClass"/></xsl:attribute>

```

```

<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">managerentry</xsl:attribute>
<xsl:attribute name="destination">intermediate</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>
</ldap:user>
</ldap:intermediateResult>

<ldap:user>
...
<xsl:attribute name="name">name</xsl:attribute>
...
<ldap:resultObject>
<xsl:attribute name="objectclass"><xsl:value-of select="$DefaultPersonClass"/></xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">name</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</ldap:resultAttribute>
</ldap:resultObject>

</ldap:user>
</ldap:StaffQueries>

```

## 示例: PersonSearch

将搜索属性更改为 MyAttribute, 将对象类更改为 mypersonclass, 并将返回属性的源更改为 myuid.

```

<ldap:StaffQueries>
...
<ldap:search>
<xsl:attribute name="filter">
(&
...
<xsl:if test="staff:parameter[@id='MyAttribute']!="">
(<xsl:value-of select="$PS_UserID"/>=
<xsl:value-of select=staff:parameter[@id='UserID']"/>)
)
</xsl:if>
...
</xsl:attribute>

<ldap:attribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>
</ldap:attribute>
...
<ldap:resultObject>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<ldap:resultAttribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</ldap:resultAttribute>
<ldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>

```

```

</sldap:resultAttribute>
</sldap:resultObject>
</sldap:search>
</sldap:StaffQueries>

```

## 示例: Users

将返回属性的源更改为 myuid, 并将对象类更改为 mypersonclass。

```

<sldap:user>
...
<xsl:attribute name="attribute">myuid</xsl:attribute>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>

<sldap:resultObject>
<xsl:attribute name="objectclass">mypersonclass</xsl:attribute>
<xsl:attribute name="usage">simple</xsl:attribute>

<sldap:resultAttribute>
<xsl:attribute name="name">myuid</xsl:attribute>
<xsl:attribute name="destination">userID</xsl:attribute>
</sldap:resultAttribute>
<sldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultMailAttribute"/></xsl:attribute>
<xsl:attribute name="destination">eMailAddress</xsl:attribute>
</sldap:resultAttribute>
<sldap:resultAttribute>
<xsl:attribute name="name"><xsl:value-of select="$DefaultLocaleAttribute"/></xsl:attribute>
<xsl:attribute name="destination">preferredLocale</xsl:attribute>
</sldap:resultAttribute>
</sldap:resultObject>

</sldap:user>

```

---

## 配置人员替代功能

通过本任务创建并激活虚拟成员管理器 (VMM) 属性扩展存储库, 以使业务流程编排器支持用户替代功能。

### 开始之前

已对联合存储库配置 WebSphere 安全性, 并且, 如果引入了定制人员分配条件, 那么还执行了第 173 页的『配置虚拟成员管理器人员目录提供程序』。您已确定是使用文件注册表、属性扩展注册表还是现有的 LDAP 模式来存储属性扩展。

### 过程

1. 在 VMM 的 PersonAccount 定义中添加两个属性, 即单值字符串『isAbsent』和多值字符串『substitutes』:
  - a. 找到 wimxmlextension.xml 文件:
    - 在 Linux、UNIX 和 i5/OS 平台上, 此文件在 *profile\_root/config/cells/cell\_name/wim/model* 目录中。
    - 在 Windows 平台上, 此文件在 *profile\_root\config\cells\cell\_name\wim\model* 目录中。
  - b. 创建 wimxmlextension.xml 文件的备份副本。
  - c. 编辑 wimxmlextension.xml 文件的原始副本并确保它包含下列定义, 这些定义将执行用户替代所需的两个属性添加到 PersonAccount 实体类型:

```

<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
 dataType="STRING" multiValued="false" propertyName="isAbsent">
 <wim:applicableEntityTypeNames>PersonAccount
 </wim:applicableEntityTypeNames>
</wim:propertySchema>

```

```

<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
 dataType="STRING" multiValued="true" propertyName="substitutes">
 <wim:applicableEntityTypeNames>PersonAccount
</wim:applicableEntityTypeNames>
</wim:propertySchema>

```

如果正在使用文件注册表 fileRegistry.xml，那么请跳转到第 182 页的 4 步骤。

2. 如果使用属性扩展注册表来存储替代信息，请执行下列操作。有关设置属性扩展存储库的更多信息，请参阅在联合存储库配置中配置属性扩展存储库。
  - a. 确保有可用于存储属性扩展的数据库。
  - b. 确保 JDBC 驱动程序类在服务器类路径中。单击环境 → WebSphere 变量以进行检查。必要时，通过单击应用程序服务器 → server\_name → 进程定义 → Java 虚拟机 → 配置将 JDBC 驱动程序添加到类路径。对于 DB2 而言，请将 db2jcc.jar,db2jcc\_license\_cu.jar 和 db2jcc\_license\_cisuz.jar 添加到服务器的类路径，然后单击应用 → 保存。
  - c. 通过使用管理控制台，为 VMM 配置 DB2 通用 JDBC 驱动程序提供程序和 4 类数据源。将数据源的 webSphereDefaultIsolationLevel 定制属性设置为值 2。有关更改缺省隔离级别的更多信息，请参阅更改非 CMP 应用程序的缺省隔离级别以及描述如何使用新定制属性 webSphereDefaultIsolationLevel 完成此更改。
  - d. 重新启动服务器。
  - e. 创建 wimlaproperties.xml 文件的备份副本。
    - 在 Linux、UNIX 和 i5/OS 平台上，此文件在 profile\_root/config/cells/cell\_name/wim/model 目录中。
    - 在 Windows 平台上，此文件在 profile\_root\config\cells\cell\_name\wim\model 目录中。
  - f. 编辑 wimlaproperties.xml 文件的原始副本，并添加下列定义：

```

<wimprop:property wimPropertyName="isAbsent" dataType="String"
 valueLength="128" multiValued="false">
 <wimprop:applicableEntityName>
 <wimprop:entityName>PersonAccount</wimprop:entityName>
 </wimprop:applicableEntityName>
</wimprop:property>

<wimprop:property wimPropertyName="substitutes" dataType="String"
 valueLength="128" multiValued="true">
 <wimprop:applicableEntityName>
 <wimprop:entityName>PersonAccount</wimprop:entityName>
 </wimprop:applicableEntityName>
</wimprop:property>

```

- g. 确保应用程序服务器处于运行状态（在 Network Deployment 环境中，确保 Deployment Manager 处于运行状态）。注意，请不要使用 wsadmin 实用程序的 -conntype NONE 选项。
- h. 使用 VMM 管理任务 setupIdMgrPropertyExtensionRepositoryTables 在属性扩展存储库数据库中创建替代属性。有关更多详细信息，请参阅使用 wsadmin 命令来设置条目映射存储库、属性扩展存储库或定制注册表数据库存储库。例如，在 Windows 平台上使用 DB2 数据库：

```

$AdminTask setupIdMgrPropertyExtensionRepositoryTables {
-reportSqlError true
-schemaLocation install_root\etc\wim\setup
-laPropXML install_root\etc\wim\setup\wimlaproperties.xml
-databaseType db2
-dbURL jdbc:db2:
-dbDriver com.ibm.db2.jcc.DB2Driver
-dbAdminId userID
-dbAdminPassword password }

```

- i. 如果正在使用轻量级目录访问协议 (LDAP) 用户存储库, 那么请编辑所使用单元的 win 子目录中 config 子目录中的 wimconfig.xml 文件并添加下列条目, 以便从 LDAP 存储库中排除替代属性:

```
<config:repositories xsi:type="config:LdapRepositoryType"
 adapterClassName="com.ibm.ws.wim.adapter.ldap.LdapAdapter"
 id="ldaprepo1" ...>
...
 <config:attributeConfiguration>
 <config:propertiesNotSupported name="isAbsent"/>
 <config:propertiesNotSupported name="substitutes"/>
 </config:attributeConfiguration>
```

- j. 激活扩展属性存储库:

- 1) 使用 setIdMgrPropertyExtensionRepository 命令。有关更多详细信息, 请参阅使用 wsadmin 命令来设置条目映射存储库、属性扩展存储库或定制注册表数据库存储库。例如, 使用 DB2 数据库 VMMDB 和数据源 VMMDS:

```
$AdminTask setIdMgrPropertyExtensionRepository {
-dataSourceName jdbc/VMMDS
-databaseType db2
-dbURL jdbc:db2:VMMDB
-dbAdminId userID
-dbAdminPassword password
-JDBCdriverClass com.ibm.db2.jcc.DB2Driver
-entityRetrievalLimit 10 }
```

- 2) 验证 wimconfig.xml 文件是否包含类似于以下的条目:

```
<config:propertyExtensionRepository
 adapterClassName="com.ibm.ws.wim.lookaside.LookasideAdapter"
 id="LA"
 databaseType="db2"
 dataSourceName="jdbc/VMMDS"
 dbAdminId="userID"
 dbAdminPassword="{xor}PasswordXOR"
 dbURL="jdbc:db2:VMMDB"
 entityRetrievalLimit="10"
 JDBCdriverClass="com.ibm.db2.jcc.DB2Driver"/>
```

3. 如果使用 LDAP 模式来存储替代信息: 可能已存在也可能尚未存在『isAbsent』和『substitutes』(名称可能不同)的定义。无论是使用现有的定义还是创建新定义, 请确保:
  - a. LDAP 目录必须允许执行写操作。
  - b. 离开信息的属性 (『isAbsent』) 必须具有布尔或字符串类型。
  - c. 用于定义被替代者的属性 (『substitutes』) 必须具有字符串类型, 接受多个值并限长 128 个字符。
  - d. 如果现有的或选择的属性名称不是『isAbsent』和『substitutes』, 那么必须在管理控制台中定义属性名称, 操作过程如下: 单击服务器 → 应用程序服务器 → **server\_name** 或服务器 → 集群 → **cluster\_name**, 接着在业务集成中展开业务流程编排器并单击人员任务管理器 → 配置 → 定制属性, 然后设置定制属性 Substitution.SubstitutesAttribute 和 Substitution.AbsenceAttribute 的期望名称。
4. 重新启动服务器。
5. 在人员任务管理器中启用替代功能:
  - a. 在管理控制台中, 单击服务器 → 应用程序服务器 → **server\_name** 或服务器 → 集群 → **cluster\_name**, 接着在业务集成中展开业务流程编排器, 然后单击人员任务管理器 → 运行时。
  - b. 要启用替代功能, 请选中启用替代。
  - c. 如果允许非管理员为其他用户执行替代, 那么清除将替代管理限制为管理员选项。

注：此设置不会影响用户更改他们自己的替代者的能力。

- d. 单击应用。
6. 如果任何步骤发生问题，请参阅 PDF 文档 *Troubleshooting WebSphere Process Server*。

## 结果

人员分配服务已配置为支持对不在场用户进行用户替代。

---

## 创建业务流程编排器的队列管理器和队列

本节描述如何创建 WebSphere MQ 队列管理器和队列。

### 开始之前

必须已安装 WebSphere MQ。

注：不推荐使用 WebSphere MQ 支持。

### 关于此任务

如果使用 WebSphere MQ 作为外部 Java 消息服务（JMS）提供程序，那么必须创建队列管理器和队列。

### 过程

1. 可选：如果正在创建生产系统，那么规划队列管理器要使用的磁盘驱动器。使用持久队列数据和 WebSphere MQ 日志的缺省位置会对队列管理器的性能产生负面影响。请考虑按照 WebSphere MQ 文档的建议更改这些位置。
2. 如果所创建的不是 WebSphere MQ 集群设置，那么执行下列操作：
  - a. 确保您的用户标识有权创建 WebSphere MQ 队列。
  - b. 创建队列管理器和队列：在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\configcreateQueues.bat queueManager
```

在 UNIX 和 Linux 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/configcreateQueues.sh queueManager
```

其中 *queueManager* 是现有队列管理器的名称或者对新队列管理器指定的名称。如果指定的队列管理器已存在，那么将使用该队列管理器来创建队列。如果该队列管理器不存在，那么将先创建并启动该队列管理器，然后再创建缺省队列。

3. 如果正在创建使用 WebSphere MQ 集群的 WebSphere 集群设置，那么只需创建集群式队列管理器和队列。
4. 如果正在创建使用中央队列管理器的 WebSphere 集群设置，那么执行下列操作：
  - a. 将 WebSphere Process Server 所在服务器上 ProcessChoreographer 目录中 config 子目录中的创建队列脚本文件复制到中央队列管理器所在的服务器：
    - 如果中央队列管理器在 Windows 工作站上，那么复制以下文件：  
createQueues.bat



- 如果中央队列管理器在 UNIX 或 Linux 服务器上，那么复制以下文件：  
`createQueues.sh`
- b. 在队列管理器所在的服务器上，确保已安装 WebSphere MQ，并确保您的用户标识有权创建 WebSphere MQ 队列。
- c. 创建队列管理器和队列： 在 Windows 平台上，输入以下命令：  
`cd install_root\ProcessChoreographer\configcreateQueues.bat queueManager`  
  
在 Linux 和 UNIX 平台上，输入以下命令：  
`cd install_root/ProcessChoreographer/configcreateQueues.sh queueManager`  
  
其中 `queueManager` 是对新队列管理器指定的名称。
- d. 通过输入以下命令，为新队列管理器添加侦听器：  
`runmqtsr -t tcp -p port -m queueManager`  
  
其中 `port` 是侦听器所侦听的端口。

## 结果

该队列管理器和队列已存在。

## 为业务流程编排器创建集群式队列管理器和队列

### 关于此任务

如果正在创建使用 WebSphere MQ 集群的业务流程编排器 WebSphere 集群设置，那么必须创建队列管理器、队列、集群、存储库、通道和侦听器。

### 过程

1. 如果 WebSphere 集群由 UNIX 节点组成，那么请对每个节点执行下列操作：
  - a. 确保您的用户标识有权创建 WebSphere MQ 队列。
  - b. 创建 GET 和 PUT 队列管理器，使它们成为 WebSphere MQ 集群的成员，并通过输入下列命令来创建队列：

```
cd install_root/ProcessChoreographer/config
createQueues.sh getQueueManager clusterName putQueueManager
```

其中：

`getQueueManager`

对 GET 队列管理器指定的唯一名称。此队列管理器主管所有本地队列。

`clusterName`

所有队列管理器所属 WebSphere MQ 集群的名称。

`putQueueManager`

PUT 队列管理器的唯一名称。此队列管理器不主管队列，这确保在所有 GET 队列之间分发消息。

如果这些队列管理器已存在，那么将使用它们。如果这些队列管理器尚未存在，那么将创建并使用它们。

- c. 通过输入以下命令，启动 WebSphere MQ 命令处理器：  
`runmqsc getQueueManager`
- d. 对于复杂的设置，建议通过输入以下 MQ 命令允许以远程方式管理队列管理器：

```
DEFINE CHANNEL('SYSTEM.ADMIN.SVRCONN') TYPE(CHLTYPE)
```

- e. 如果此队列管理器将成为 WebSphere MQ 集群的存储库，那么请输入以下 MQ 命令：

```
ALTER QMGR REPOS('clusterName') REPOSNL(' ')
```

- f. 通过输入下列 MQ 命令，为此服务器上未主管的每个存储库定义队列管理器发送方通道和接收方通道。对于每个集群接收方通道：

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
 CHLTYPE(CLUSRCVR) +
 CLUSTER('clusterName') +
 CLUSNL(' ') +
 CONNAME('repositoryIP-Address(port)') +
 DESCR('Cluster receiver channel at repositoryQueueManager TCP/IP') +
 MAXMSGL(4194304) +
 TRPTYPE(TCP) +
 MCAUSER('principal') +
 REPLACE
```

对于每个集群发送方通道：

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
 CHLTYPE(CLUSSDR) +
 CONNAME('repositoryIP-Address(port)') +
 CLUSTER('clusterName') +
 CLUSNL(' ') +
 DESCR('Cluster sender channel to repositoryQueueManager TCP/IP') +
 MAXMSGL(4194304) +
 TRPTYPE(TCP) +
 MCAUSER('targetPrincipal') +
 REPLACE +
 NPMSPEED (NORMAL)
```

其中：

*repositoryQueueManager*

主管存储库的队列管理器的名称。

*clusterName*

所有队列管理器所属 WebSphere MQ 集群的名称。

*repositoryIP-Address*

存储库队列管理器所在节点的 IP 地址。

*port* 存储库队列管理器正在使用的 IP 端口。

*principal, targetPrincipal*

接收通道和发送通道使用的 MCAUSER。有关此值的更多信息，请参阅 WebSphere MQ 文档。

- g. 对于每个队列管理器，通过输入以下 MQ 命令来启动侦听器：

```
runmqtsr -t tcp -p port -m QueueManager
```

2. 如果 WebSphere 集群由 Windows 节点组成，那么请对每个节点执行下列操作：

- a. 确保您的用户标识有权创建 WebSphere MQ 队列。
- b. 创建“GET”队列管理器，使其成为 WebSphere MQ 集群的成员，并通过输入下列命令创建队列：

```
cd install_root\ProcessChoreographer\config
createQueues.bat getQueueManager clusterName putQueueManager
```

其中：

*getQueueManager*

对 GET 队列管理器指定的唯一名称。此队列管理器主管所有本地队列。

*clusterName*

所有队列管理器所属 WebSphere MQ 集群的名称。

*putQueueManager*

PUT 队列管理器的唯一名称。此队列管理器不主管队列，这确保在所有 GET 队列之间分发消息。

如果这些队列已存在，那么将使用它们。如果这些队列尚未存在，那么将创建并使用它们。

- c. 通过输入以下命令，启动 WebSphere MQ 命令处理器：

```
runmqsc queueManager
```

- d. 对于复杂的设置，建议通过输入以下 MQ 命令允许以远程方式管理队列管理器：

```
DEFINE CHANNEL('SYSTEM.ADMIN.SVRCONN') TYPE(CHLTYPE)
```

- e. 如果此队列管理器将成为 WebSphere MQ 集群的存储库，那么请输入以下 MQ 命令：

```
ALTER QMGR REPOS('clusterName') REPOSNL(' ')
```

- f. 通过输入下列 MQ 命令，为此服务器上未主管的每个存储库定义队列管理器发送方通道和接收方通道。对于每个集群接收方通道：

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
 CHLTYPE(CLUSRCVR) +
 CLUSTER('clusterName') +
 CLUSNL(' ') +
 CONNAME('repositoryIP-Address(port)') +
 DESCR('Cluster receiver channel at repositoryQueueManager TCP/IP') +
 MAXMSGL(4194304) +
 TRPTYPE(TCP) +
 MCAUSER('principal') +
 REPLACE
```

对于每个集群发送方通道：

```
DEFINE CHANNEL('TO.repositoryQueueManager.TCP') +
 CHLTYPE(CLUSSDR) +
 CONNAME('repositoryIP-Address(port)') +
 CLUSTER('clusterName') +
 CLUSNL(' ') +
 DESCR('Cluster sender channel to repositoryQueueManager TCP/IP') +
 MAXMSGL(4194304) +
 TRPTYPE(TCP) +
 MCAUSER('principal') +
 REPLACE +
 NPMSPEED (NORMAL)
```

其中：

*repositoryQueueManager*

主管存储库的队列管理器的名称。

*clusterName*

所有队列管理器所属 WebSphere MQ 集群的名称。

*repositoryIP-Address*

存储库队列管理器所在节点的 IP 地址。

*port* 存储库队列管理器正在使用的 IP 端口。

*principal*

要使用的 MCAUSER。有关此值的更多信息，请参阅 WebSphere MQ 文档。

- g. 对于每个队列管理器，通过输入以下 MQ 命令来启动侦听器：

```
runmqtsr -t tcp -p port -m QueueManager
```

3. 可选: 要验证服务器上的通道状态, 请输入以下 MQ 命令:
- ```
display chstatus(*)
```

结果

队列管理器、队列、集群、存储库、通道和侦听器已存在。

概述: 配置业务流程编排器资源管理器

关于此任务

业务流程编排器资源管理器提供了用于管理业务流程和处理人员任务的用户界面。业务流程编排器资源管理器是一个 Java 2 Enterprise Edition (J2EE) Web 应用程序, 并且基于 JavaServer Faces (JSF) 技术和业务流程编排器资源管理器组件。

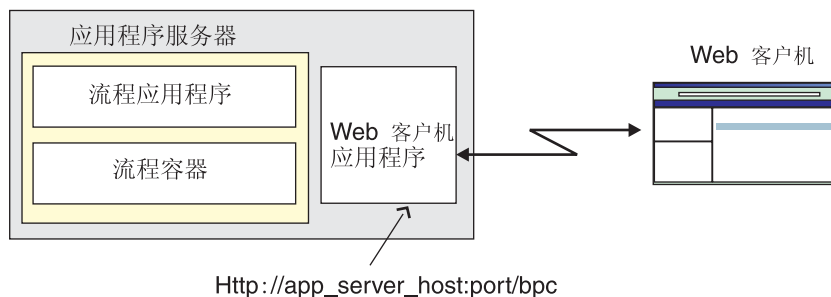
- 第 111 页的『关于业务流程编排器资源管理器』
- 第 188 页的『配置业务流程编排器资源管理器』

关于业务流程编排器资源管理器

业务流程编排器资源管理器是一个 Web 应用程序, 它实现了用于与业务流程和人员任务进行交互的通用 Web 用户界面。

可以在服务器或集群中配置一个或多个业务流程编排器资源管理器实例。使用带有 WebSphere Process Server 概要文件的 WebSphere Process Server 安装或者 WebSphere Process Server 客户机安装已足够 - 不必在服务器或集群中配置业务流程编排器。客户机安装是将客户机连接至 WebSphere Process Server 时所需的唯一基础结构, 但它未包含业务流程编排器资源管理器。但是, 如果有 Deployment Manager, 那么还可以将业务流程编排器资源管理器安装在 WebSphere Process Server 客户机安装中的服务器上。

一个业务流程编排器资源管理器只能连接至一个业务流程编排器配置, 但不必连接至本地配置。但是, 可以在同一服务器或集群中配置多个业务流程编排器资源管理器实例, 并且每个实例都可以连接至不同的业务流程编排器配置。



启动业务流程编排器资源管理器之后, 用户界面中显示的对象以及可以执行的操作取决于您所属的用户组以及该组被授予的权限。例如, 如果您是业务流程管理员, 那么对所部署的业务流程的平稳运行负责。您可以查看有关流程模板、任务模板、流程实例、任务实例及其相关信息。您还可以对这些对象执行操作; 例如, 可以启动

新的流程实例、创建和启动任务、修复和重新启动失败的活动、管理工作项以及删除已完成的流程实例和任务实例。但是，如果您是用户，那么只能查看那些分配给您的任务并对它们执行操作。

配置业务流程编排器资源管理器

可以通过运行脚本或使用管理控制台配置业务流程编排器资源管理器。

开始之前

已配置业务流程编排器。

关于此任务

下列其中一种或多种情况适用：

- 尚未安装业务流程编排器资源管理器。
- 要管理现有的业务流程编排器配置。
- 要将另一个业务流程编排器资源管理器实例添加到已管理的业务流程编排器配置。

要配置业务流程编排器资源管理器，请执行下列其中一项操作：

- 如果要使用脚本，请执行第 189 页的『使用 clientconfig.jacl 脚本文件配置业务流程编排器资源管理器』。
- 如果要使用管理控制台，请执行『使用管理控制台配置业务流程编排器资源管理器』。

结果

业务流程编排器资源管理器已配置完毕并可供使用。

下一步做什么？

启动业务流程编排器资源管理器。

使用管理控制台配置业务流程编排器资源管理器

可以使用管理控制台来配置业务流程编排器资源管理器。

过程

1. 单击**服务器** → **应用程序服务器** → **server_name** 或**服务器** → **集群** → **cluster_name**，然后在**业务集成**中展开**业务流程编排器**并单击**业务流程编排器资源管理器**。
2. 要创建新的资源管理器配置，请单击**添加**。
3. 输入下列字段的值：
 - 上下文根在部署目标服务器或集群中必须唯一。
 - 资源管理器搜索结果限制。
 - 受管业务流程编排器容器

有关这些字段的更多信息，请参阅第 123 页的『业务流程编排器资源管理器设置』。

4. 单击**应用**。系统将显示消息以指示进度。
5. 可选：如果报告了任何问题，请检查 SystemOut.log 文件。

6. 启动名为 `BPCExplorer_scope` 的企业应用程序。其中, `scope` 指定要在其中配置业务流程编排器资源管理器的服务器或集群。

结果

业务流程编排器资源管理器已配置完毕并可供使用。

下一步做什么?

启动业务流程编排器资源管理器。

使用 `clientconfig.jacl` 脚本文件配置业务流程编排器资源管理器

此脚本文件在服务器或集群中配置业务流程编排器资源管理器以及所有必需的资源。

用途

此脚本文件配置业务流程编排器资源管理器。此脚本文件能够以交互方式或批处理方式运行。

位置

`clientconfig.jacl` 脚本文件在业务流程编排器的 `config` 目录中:

- 对于 Linux、UNIX 和 i5/OS 平台: 在 `install_root/ProcessChoreographer/config` 目录中
- 对于 Windows 平台: 在 `install_root\ProcessChoreographer\config` 目录中

在独立服务器环境中运行此脚本

在独立服务器环境中:

- 仅当应用程序服务器未处于运行状态时, 才需要指定 `-conntype NONE` 选项。
- 如果服务器处于运行状态, 并且已启用 WebSphere 管理安全性, 那么请指定 `-user` 和 `-password` 选项。
- 如果未配置缺省概要文件, 那么请添加 `-profileName` 选项。

在 **Network Deployment** 环境中运行此脚本

在 Network Deployment 环境中:

- 在 Deployment Manager 节点上运行此脚本。
- 仅当 Deployment Manager 未处于运行状态时, 才需要指定 `-conntype NONE` 选项。
- 如果已启用 WebSphere 管理安全性, 那么请指定 `-user` 和 `-password` 选项。
- 如果未配置缺省概要文件, 那么请添加 `-profileName` 选项。

以非交互方式配置业务流程编排器资源管理器

将当前目录切换到 `install_root` 并执行下列操作:

在 Linux 和 UNIX 平台上, 输入以下命令:

```
bin/wsadmin.sh -f ProcessChoreographer/config/clientconfig.jacl options
```

在 i5/OS 平台上, 输入以下命令:

```
bin/wsadmin -f ProcessChoreographer/config/clientconfig.jacl options
```

在 Windows 平台上，输入以下命令：

```
bin\wsadmin.bat -f ProcessChoreographer/config/clientconfig.jacl options
```

其中，*options* 是：

```
( [-user userName][-password password] | [-conntype NONE] )
  [-profileName profileName]
( [-node nodeName][-server serverName] )
  [-cluster clusterName]
  [-contextRootExplorer explorerContextRoot]
  [-hostName explorerVirtualHostname]
  [-precompileJSPs { yes | no }]
( ( [-remoteNode nodeName][-remoteServer serverName] )
  | [-remoteCluster clusterName] )
  [-maxListEntries maximum]
  [-explorerHost explorerURL]
```

注：如果在命令行上提供了所有必需参数，那么不会提示您输入这些参数。对于命令行中未指定的任何必需参数，系统将按它们的列示顺序以交互方式提示您进行输入。

参数

使用 `wsadmin` 调用此脚本时，可以指定下列参数：

-node *nodeName*

其中，*nodeName* 是要配置业务流程编排器资源管理器的节点的名称。如果未指定此参数，那么缺省值是本地节点。

-server *serverName*

其中 *serverName* 是要配置业务流程编排器资源管理器的服务器的名称。如果只有一个节点，并且只有一个服务器，那么此参数是可选的。

-cluster *clusterName*

其中，*clusterName* 是要配置业务流程编排器资源管理器的集群的名称。此参数是可选的。在独立服务器环境中，请不要指定此选项。如果指定了节点和服务器，也不要指定此选项。

-contextRootExplorer *contextRootExplorer*

其中 *contextRootExplorer* 是业务流程编排器资源管理器的上下文根。上下文根在 WebSphere 单元中必须唯一。缺省值是 `/bpc`。

-hostName *VirtualHostname*

其中，*VirtualHostname* 是将要运行业务流程编排器以及业务流程管理器 API 和人员任务管理器 API 的 Web Service 绑定的虚拟主机。缺省值是 `default_host`。

-precompileJSPs { no | yes }

确定是否对 Java Server Pages (JSP) 进行预编译。缺省值为 `no`。请注意，不可能对预编译的 JSP 进行调试。

-remoteNode *nodeName*

如果您不想连接至本地业务流程编排器配置，那么请使用此参数和 `remoteServer`。如果未指定此参数，那么缺省值为 `-node` 参数值。

-remoteServer *serverName*

如果您不想连接至本地业务流程编排器配置，那么请使用此参数和 `remoteNode`。如果未指定此参数，那么缺省值为 `-server` 参数值。

-remoteCluster *clusterName*

如果您不想连接至本地业务流程编排器配置，并且未指定 `remoteNode` 和 `remoteServer`，请使用此参数。如果未指定此参数，那么缺省值为 `-cluster` 参数值。

-maxListEntries *maximum*

其中，*maximum* 是业务流程编排器资源管理器对一个查询返回的最大结果数。缺省值是 10000。

-explorerHost *explorerURL*

其中 *explorerURL* 是业务流程编排器资源管理器的 URL。此参数值用于将所管理的业务流程编排器配置的人员任务管理器链接至此特定业务流程编排器资源管理器实例。此参数的缺省值为空字符串，这表示不建立该链接。以后，您可以使用管理控制台来建立或更改该链接。

日志文件

如果使用 `clientconfig.jacl` 脚本文件创建配置时发生了问题，那么请检查下列日志文件：

- `clientconfig.log`
- `wsadmin.traceout`

这两个文件都在概要文件的 `logs` 目录中：

- 对于 Linux、UNIX 和 i5/OS 平台：在 `profile_root/logs` 目录中
- 对于 Windows 平台：在 `profile_root\logs` 目录中

如果以连接方式运行此脚本，那么还需检查 `SystemOut.log` 和 `SystemErr.log` 文件。这两个文件在 `logs` 目录下与 `wsadmin` 脚本编制客户机所连接的应用程序服务器或 Deployment Manager 同名的子目录中。

配置业务流程编排器观察器

您并非必须使用业务流程编排器观察器，但是，在可以使用它之前，必须设置数据库并安装应用程序。

开始之前

已执行第 112 页的『关于业务流程编排器观察器』。

关于此任务

您希望在业务流程编排器观察器具有自己的数据库的情况下对其进行配置。

过程

1. 执行第 193 页的『为业务流程编排器观察器准备数据库』。
2. 执行第 224 页的『配置业务流程编排器事件收集器应用程序』。
3. 执行第 230 页的『配置业务流程编排器观察器应用程序』。
4. 执行第 234 页的『更改业务流程编排器观察器的配置参数』。
5. 执行第 232 页的『对业务流程编排器启用日志记录』。
6. 执行第 242 页的『验证业务流程编排器观察器』。

结果

业务流程编排器观察器已配置完毕并运行正常。

您可以使用业务流程编排器观察器来生成报告，如第 325 页的『创建业务流程报告和活动报告』所述。

除去业务流程编排器观察器 V6.0.1 样本

本节描述如何除去 V6.0.1 附带提供的业务流程编排器观察器样本。

关于此任务

如果已从 WebSphere Process Server V6.0.1 升级，并且曾使用业务流程编排器观察器样本，那么必须先除去该样本，然后才能配置最新版本的业务流程编排器观察器。应用程序本身不会被迁移，但您必须删除数据库视图和索引。此样本所收集的数据不会被迁移，并且不可用。

过程

删除业务流程编排器观察器和事件收集器所使用的数据库视图和索引。

1. 如果使用 DB2 数据库:

- a. 连接至观察器模式所在的数据库。
- b. 找到并运行用于删除观察器样本 DB2 模式的脚本:
 - 在 Windows 平台上: `install_root\ProcessChoreographer\sample\observer\dropObserverSampleSchema_DB2.sql`。
 - 在 Linux 和 UNIX 平台上: `install_root/ProcessChoreographer/sample/observer/dropObserverSampleSchema_DB2.sql`。

例如，输入以下命令:

```
db2 -tf dropObserverSampleSchema_DB2.sql
```

2. 如果已使用 Cloudscape™ 数据库，那么请在命令窗口中执行下列操作:

- a. 将 `derby.jar` 和 `derbytools.jar` 文件添加至 `CLASSPATH`。
 - 在 Windows 平台上: 这些文件在 `install_root\derby\bin\embedded` 目录中。
 - 在 Linux 和 UNIX 平台上: 这些文件在 `install_root/derby/bin/embedded` 目录中。
- b. 找到用于删除观察器样本 Cloudscape 模式的脚本，然后按照该脚本文件中注释所描述的方式运行该脚本:
 - 在 Windows 平台上: `install_root\ProcessChoreographer\sample\observer\dropObserverSampleSchema_Cloudscape.sql`。
 - 在 Linux 和 UNIX 平台上: `install_root/ProcessChoreographer/sample/observer/dropObserverSampleSchema_Cloudscape.sql`。

例如，输入以下命令:

```
java -Dij.protocol=jdbc:derby:  
-Dij.database=OBSVRDB  
org.apache.derby.tools.ij  
dropObserverSampleSchema_Cloudscape.sql
```

结果

已除去业务流程编排器观察器样本。

为业务流程编排器观察器准备数据库

执行数据库操作。

为业务流程编排器观察器准备 DB2 通用数据库

可以使用脚本或交互式工具来准备数据库。

使用 SQL 脚本为业务流程编排器观察器准备 DB2 通用数据库:

本节描述如何在 Linux、UNIX 和 Windows 平台上使用 createTablespace_Observer.sql 和 createSchema_Observer.sql 脚本来准备 DB2 通用数据库。

关于此任务

该数据库必须已存在。您可以使用现有数据库，也可以根据数据库文档创建新数据库。要执行此任务，您必须具有目标数据库的管理权。

过程

1. 切换到数据库的配置脚本所在的业务流程编排器子目录。
 - 在 Linux 和 UNIX 平台上，切换到 `install_root/dbscripts/ProcessChoreographer/DB2` 目录。
 - 在 Windows 平台上，切换到 `install_root\dbscripts\ProcessChoreographer\DB2` 目录。
2. 将所有 *Observer.sql 脚本文件复制至数据库服务器。
3. 在数据库服务器上，切换到已复制的脚本文件所在的目录。
4. 创建表空间:
 - a. 按照 createTablespace_Observer.sql 脚本文件开头的指示信息，编辑此文件。
 - b. 运行表空间创建脚本文件，输入以下命令：

```
db2 -tf createTablespace_Observer.sql
```
 - c. 确保脚本输出未包含任何错误。如果发生了错误，那么可以使用 dropTablespace_Observer.sql 脚本文件删除该表空间。
5. 创建模式（表、索引和视图）。
 - a. 按照 createSchema_Observer.sql 脚本文件开头的指示信息，编辑此文件。
 - b. 在 DB2 命令行处理器中，输入以下命令：

```
db2 -tf createSchema_Observer.sql
```
 - c. 确保脚本输出未包含任何错误。如果要删除该模式，请使用 dropSchema_Observer.sql 脚本文件。
6. 如果要使用业务流程编排器观察器 UDF 的 Java 实现来代替 SQL 实现，请执行第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』。
7. 使用管理控制台来创建指向该数据库的 XA 数据源，然后测试连接。

结果

业务流程编排器观察器的数据库模式已准备完毕。

相关概念

第 222 页的『业务流程编排器观察器的用户定义函数』

借助业务流程编排器观察器，可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告，业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数（UDF）。

相关任务

第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』

通过使用 `setupEventCollector` 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数（UDF）与基于 SQL 的 UDF 之间进行切换。

使用 `setupEventCollector` 工具为业务流程编排器观察器准备 DB2 通用数据库:

本节描述如何在 Linux、UNIX 和 Windows 平台上使用菜单驱动的交互式工具和 `createTablespace_Observer.sql` 脚本来准备 DB2 通用数据库。

开始之前

该数据库必须已存在，并且您对该数据库必须具有管理权。

过程

1. 如果使用 2 类 JDBC 连接:
 - a. 准备命令行环境:
 - 在 Linux 和 UNIX 平台上，运行所使用 DB2 实例的 `db2profile`。
 - 在 Windows 上，打开 DB2 命令窗口。
 - b. 如果使用远程数据库，那么请在本地 DB2 实例上对该数据库进行编目。
2. 创建表空间:
 - a. 切换到数据库的 SQL 脚本所在的业务流程编排器子目录。
 - 在 Linux 和 UNIX 平台上，切换到 `install_root/dbscripts/ProcessChoreographer/DB2` 目录。
 - 在 Windows 平台上，切换到 `install_root\dbscripts\ProcessChoreographer\DB2` 目录。
 - b. 创建 `createTablespace_Observer.sql` 脚本文件的副本。
 - c. 按照 `createTablespace_Observer.sql` 脚本文件开头的指示信息，编辑此文件的副本。
 - d. 运行表空间创建脚本文件，输入以下命令：

```
db2 -tf createTablespace_Observer.sql
```
 - e. 确保脚本输出未包含任何错误。如果发生了错误，那么可以使用 `dropTablespace_Observer.sql` 脚本文件删除该表空间。
3. 如果未使用数据库管理员的用户标识，那么请确保该用户标识具有下列权限：

```
GRANT CREATETAB, CONNECT, CREATE_EXTERNAL_ROUTINE ON DATABASE
  TO USER user_name;
GRANT USE OF TABLESPACE tablespace_name TO USER user_name;
```

其中，`user_name` 是用户标识，`tablespace_name` 是所有观察器表空间名的列表，您可以在 `createTablespace_Observer.sql` 脚本中找到此列表。
4. 切换到配置脚本所在的业务流程编排器目录。

在 Linux 和 UNIX 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\config
```

5. 启动用于设置事件收集器的工具，如第 238 页的『setupEventCollector 工具』所述。
6. 准备数据库：

a. 当您看到：

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

- 0) Exit Menu

请选择选项 1，以便为事件收集器和观察器应用程序准备数据库。

b. 当您看到：

```
Prepare a database for the WebSphere Business Process Choreographer  
Event Collector and Observer
```

```
Select the type of your database provider:
```

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) DB2 V7 on z/OS
- 8) DB2 V8 on z/OS
- o) Oracle

- 0) Exit Menu

请输入 d，以选择 DB2 通用数据库。

- c. 此工具允许您创建 SQL 文件，接着，可以将此文件交给数据库管理员运行，而不是使用您的当前用户标识来运行此文件。当您看到：

```
Do you want to create an SQL file only (delay database preparation)?
```

- y) yes
- n) no

- 如果您不想推迟运行 SQL，那么请输入 n。
- 如果您想推迟运行 SQL，那么请输入 y。您将看到：

```
Even if you want to delay the configuration,  
your entered values can be checked within the database.
```

```
Do you want to perform these checks?
```

- y) yes
- n) no

- 如果您希望在数据库中对输入的值进行检查，那么请输入 y。
- 否则，请输入 n。

根据所输入内容的不同，您可能不会看到下列所有提示。请跳过您未看到的步骤。

d. 如果您看到：

```
Specify the JDBC driver type to be used:
```

- 2) Connect using type 2 (using a native database client)
- 4) Connect using type 4 (directly via JDBC)

请指定 JDBC 驱动程序类型：

- 如果正在使用本机数据库客户机，请输入 2。
 - 否则，请输入 4 以选择 4 类 JDBC 驱动程序。
- e. 如果您看到下列其中一个提示:
- ```
Specify the name of your database: [BPEDB]
Specify the name of database in local catalog: [BPEDB]
```
- 输入数据库名称或别名。
- 注:** 缺省值 BPEDB 就是业务流程编排器所使用的数据库。对于高性能系统而言，应该使用另一个数据库。
- f. 如果您看到:
- ```
Specify the hostname of the database server: [localhost]
```
- 请输入数据库服务器的主机名或 IP 地址。
- g. 如果您看到:
- ```
Specify the port where the database server is listening: [50000]
```
- 请输入数据库服务器的端口号。
- h.
- ```
Specify the directory of your JDBC driver: [D:\opt\SQLLIB\java]
```
- 请输入 JDBC 驱动程序的 JAR 文件 db2jcc.jar 和 db2jcc_license_cu.jar 的所在目录。
- i. 如果您看到:
- ```
Specify userid to connect to the database 'database_name' [db2admin] :
Specify the password for userid 'user_ID' :
```
- 请输入用于连接至数据库的用户标识和密码。将显示以下内容:
- ```
Trying to connect to database 'database_name', using user 'user_ID'
Connected to 'database_name'
```
- j. 如果您看到:
- ```
Specify the database schema to be used. [user_ID] :
```
- 请输入数据库对象所使用的数据库模式（集合名）。如果输入了空格字符或者未填写此字段，那么将使用该用户标识的模式。
- k. 如果您看到:
- ```
Choose the implementation of the Observer user-defined functions.

Note: The Java UDFs are more precise, but they require a jar file
installed to the database.
Visit the Observer documentation for details.

1) Java
2) SQL

0) Exit Menu
```
- 如果要使用较为精确的基于 Java 的用户定义函数（UDF），那么请输入 1，这将要求在数据库中安装一个 JAR 文件。
 - 如果要使用不太精确的基于 SQL 的 UDF，那么请输入 2。

您将看到类似于以下的内容:

```
Checking for required tablespace(es) ['OBSVRTS']
All required tablespaces were found.
Loading the jar file 'install_root\lib\bpcodbutil.jar' into the database.
The jar file 'install_root\lib\bpcodbutil.jar' was successfully installed.
```

The setup of the database completed successfully.

7. 使用管理控制台来创建指向该数据库的 XA 数据源，然后测试连接。

结果

业务流程编排器观察器的数据库模式已准备完毕。

相关概念

第 222 页的『业务流程编排器观察器的用户定义函数』

借助业务流程编排器观察器，可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告，业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数 (UDF)。

相关任务

第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』

通过使用 setupEventCollector 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数 (UDF) 与基于 SQL 的 UDF 之间进行切换。

为业务流程编排器观察器准备 DB2 iSeries 版数据库

可以使用脚本或交互式工具来准备数据库。

使用 SQL 脚本为业务流程编排器观察器准备 DB2 iSeries 版数据库:

本节描述如何在 i5/OS qshell 环境中使用 createSchema_Observer.sql 脚本来准备 DB2 iSeries 数据库。

开始之前

集合必须已存在。您可以使用现有集合，也可以根据数据库文档创建新集合。

过程

1. 在 Qshell 环境中，切换到数据库的配置脚本所在的业务流程编排器子目录。输入以下命令：

```
cd install_root/dbscripts/ProcessChoreographer/DB2iSeries
```
2. 将所有 *Observer.sql 脚本文件复制至数据库服务器。
3. 在数据库服务器上，切换到已复制的脚本文件所在的目录。
4. 创建模式（表、索引和视图）。
 - a. 按照 createSchema_Observer.sql 脚本文件开头的指示信息，编辑此文件。
 - b. 在 DB2 命令行处理器或 Qshell 中，输入以下命令：

```
db2 -tf createSchema_Observer.sql
```
 - c. 确保脚本输出未包含任何错误。如果要删除该模式，请使用 dropSchema_Observer.sql 脚本文件。
5. 如果要使用所需用户定义函数 (UDF) 的 Java 实现，请执行第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』。
6. 使用管理控制台来创建指向该数据库的 XA 数据源，然后测试连接。

结果

业务流程编排器观察器的数据库模式已准备完毕。

相关概念

第 222 页的『业务流程编排器观察器的用户定义函数』

借助业务流程编排器观察器，可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告，业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数 (UDF)。

相关任务

第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』

通过使用 `setupEventCollector` 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数 (UDF) 与基于 SQL 的 UDF 之间进行切换。

使用 `setupEventCollector` 工具为业务流程编排器观察器准备 DB2 iSeries 版数据库:

本节描述如何从 i5/OS Qshell 环境中使用菜单驱动的交互式工具来准备 DB2 iSeries 数据库。

关于此任务

您可以使用现有集合，也可以根据数据库文档创建新集合。

过程

1. 启动 Qshell 环境。
2. 切换到配置脚本所在的业务流程编排器目录。输入以下命令：
`cd install_root/ProcessChoreographer/config`
3. 启动用于设置事件收集器的工具，如第 238 页的『`setupEventCollector` 工具』所述。
4. 准备数据库：

a. 当您看到：

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

请选择选项 1，以便为事件收集器和观察器应用程序准备数据库。

b. 当您看到：

Prepare a database for the WebSphere Business Process Choreographer
Event Collector and Observer

Select the type of your database provider:

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) DB2 V7 on z/OS

- 8) DB2 V8 on z/OS
- o) Oracle

- 0) Exit Menu

输入 i 以选择 DB2 iSeries 版。

- c. 此工具允许您创建 SQL 文件，接着，可以将此文件交给数据库管理员运行，而不是使用您的当前用户标识来运行此文件。当您看到：

```
Do you want to create an SQL file only (delay database preparation)?
y) yes
n) no
```

- 如果您不想推迟运行 SQL，那么请输入 n。
- 如果您想推迟运行 SQL，那么请输入 y。您将看到：

```
Even if you want to delay the configuration,
your entered values can be checked within the database.
Do you want to perform these checks?
y) yes
n) no
```

- 如果您希望在数据库中对输入的值进行检查，那么请输入 y。
- 否则，请输入 n。

根据所输入内容的不同，您可能不会看到下列所有提示。请跳过您未看到的步骤。

- d. 如果您看到：

Specify the JDBC driver to be used:

- 1) Connect using the IBM Toolbox for Java JDBC driver
 - 2) Connect using the native JDBC driver
- Your selection: [2]

- 如果正在配置远程数据库，那么请输入 1 以选择 IBM Toolbox for Java JDBC 驱动程序。
- 如果正在配置本地数据库，那么请输入 2 以选择本机 JDBC 驱动程序。

- e. 如果您看到：

Specify the name of database in local catalog: [*LOCAL]

或

Specify the name of your database: [*SYSBAS]

请输入服务标识或接受缺省值。

- f. 如果您看到：

Specify the hostname of the database server: [localhost]

请输入数据库服务器的主机名或 IP 地址。

- g. 如果您看到：

Specify the directory of your JDBC driver:
[/QIBM/ProdData/HTTP/Public/jt400/lib]

请输入 JDBC 驱动程序的 JAR 文件所在的目录。

- 对于本机驱动程序 (db2_classes.zip) 而言，这通常是 /QIBM/ProdData/Java400/ext。

- 对于 Toolbox 驱动程序 (jt400.jar) 而言, 这通常是 /QIBM/ProdData/HTTP/Public/jt400/lib。
- h. 如果您看到:
- ```
Specify userid to connect to the database 'database_name' [db2admin] :
Specify the password for userid 'user_ID' :
```
- 请输入用于连接至数据库的用户标识和密码。
- i. 如果您看到:
- ```
Specify the database schema to be used. [user_ID] :
```
- 请输入数据库对象所使用的数据库模式 (集合名)。必须指定已存在的模式。如果输入了空格字符或者未填写此字段, 那么将使用该用户标识的模式。
- j. 如果您看到:
- Note: The Java UDFs are more precise, but they require a jar file installed to the database.
Visit the Observer documentation for details.
- 1) Java
 - 2) SQL
 - 0) Exit Menu
- 如果要使用较为精确的基于 Java 的用户定义函数 (UDF), 那么请输入 1, 这将要求在数据库中安装一个 JAR 文件。
 - 如果要使用不太精确的基于 SQL 的 UDF, 那么请输入 2。
- k. 成功地准备数据库后, 将显示以下内容:
- ```
The setup of the database completed successfully.
```
5. 如果使用了独立的数据库, 而未使用 BPEDB, 那么请使用管理控制台来创建指向该数据库的 XA 数据源, 然后测试连接。

## 结果

业务流程编排器观察器的数据库模式已准备完毕。

### 相关概念

第 222 页的『业务流程编排器观察器的用户定义函数』

借助业务流程编排器观察器, 可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告, 业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数 (UDF)。

### 相关任务

第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』

通过使用 `setupEventCollector` 工具或运行脚本, 可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数 (UDF) 与基于 SQL 的 UDF 之间进行切换。

## 使用 `setupEventCollector` 工具从远程系统准备 DB2 iSeries 版数据库:

本节描述如何使用菜单驱动的交互式工具从远程 Linux、Windows 或 UNIX 系统为业务流程编排器观察器准备 DB2 iSeries 版数据库。

## 关于此任务

您可以使用现有集合，也可以根据数据库文档创建新集合。要使用的集合必须已存在。

## 过程

1. 要以远程方式准备数据库，需要下载用于连接至 iSeries 机器的 IBM Toolbox JDBC 驱动程序。下载完成后，请记录 JAR 文件 `jt400.jar` 的位置。
2. 启动命令行环境。
3. 切换到配置脚本所在的业务流程编排器目录。
  - 在 Windows 平台上，输入以下命令：  
`cd install_root\ProcessChoreographer\config`
  - 在 Linux 和 UNIX 平台上，输入以下命令：  
`cd install_root/ProcessChoreographer/config`
4. 启动用于设置事件收集器的工具，如第 238 页的『`setupEventCollector` 工具』所述。
5. 准备数据库：
  - a. 当您看到：

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

0) Exit Menu

请选择选项 1，以便为事件收集器和观察器应用程序准备数据库。

- b. 当您看到：

Prepare a database for the WebSphere Business Process Choreographer  
Event Collector and Observer

Select the type of your database provider:

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) DB2 V7 on z/OS
- 8) DB2 V8 on z/OS
- o) Oracle

0) Exit Menu

输入 i 以选择 DB2 iSeries 版。

- c. 此工具允许您创建 SQL 文件，接着，可以将此文件交给数据库管理员运行，而不是使用您的当前用户标识来运行此文件。当您看到：

Do you want to create an SQL file only (delay database preparation)?

- y) yes
- n) no

- 如果您不想推迟运行 SQL，那么请输入 n。
- 如果您想推迟运行 SQL，那么请输入 y。您将看到：

Even if you want to delay the configuration,  
your entered values can be checked within the database.

Do you want to perform these checks?

- y) yes
- n) no

- 如果您希望在数据库中对输入的值进行检查, 那么请输入 `y`。
- 否则, 请输入 `n`。

根据所输入内容的不同, 您可能不会看到下列所有提示。请跳过您未看到的步骤。

d. 如果您看到:

Specify the name of your database: [`*SYSBAS`]

请输入服务标识或接受缺省值。

e. 如果您看到:

Specify the hostname of the database server: [`localhost`]

请输入数据库服务器的主机名或 IP 地址。

f. 如果您看到:

Specify the directory of your JDBC driver:  
[`/QIBM/ProdData/HTTP/Public/jt400/lib`]

请输入 JDBC 驱动程序文件 `jt400.jar` 被下载到的目录。

g. 如果您看到:

Specify userid to connect to the database 'database\_name' [`db2admin`] :  
Specify the password for userid 'user\_ID' :

请输入用于连接至数据库的用户标识和密码。

h. 如果您看到:

Specify the database schema to be used. [`user_ID`] :

请输入数据库对象所使用的数据库模式 (集合名)。必须指定已存在的模式。如果输入了空格字符或者未填写此字段, 那么将使用该用户标识的模式。

i. 如果您看到:

Note: The Java UDFs are more precise, but they require a jar file installed to the database.  
Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 如果要使用较为精确的基于 Java 的用户定义函数 (UDF), 那么请输入 1, 这将要求在数据库中安装一个 JAR 文件。
- 如果要使用不太精确的基于 SQL 的 UDF, 那么请输入 2。

j. 成功地准备数据库后, 将显示以下内容:

The setup of the database completed successfully.

6. 如果使用了独立的数据库, 而未使用 `BPEDB`, 那么请使用管理控制台来创建指向该数据库的 XA 数据源, 然后测试连接。

## 结果

业务流程编排器观察器的数据库模式已准备完毕。

## 相关概念

第 222 页的『业务流程编排器观察器的用户定义函数』

借助业务流程编排器观察器，可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告，业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数（UDF）。

### 相关任务

第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』

通过使用 setupEventCollector 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数（UDF）与基于 SQL 的 UDF 之间进行切换。

## 为业务流程编排器观察器准备 DB2 z/OS 版数据库

可以通过远程方式或者在 UNIX 系统服务中准备数据库。

### 在 USS 中为业务流程编排器观察器创建 DB2 z/OS 版数据库:

本节描述如何在 z/OS 机器上的 UNIX 系统服务（USS）中使用菜单驱动的交互工具以及 createTablespace\_Observer.sql 脚本来创建 DB2 z/OS 版数据库。

### 过程

#### 1. 准备 DB2 环境:

- a. 登录到本机 z/OS 环境。
- b. 如果安装了多个 DB2 系统，那么请确定要使用的子系统。
- c. 记录 DB2 子系统所侦听的 IP 端口。
- d. 确定子系统的位置名。要确定位置名，请查看“DB2 系统”面板，或者对子系统选择 DB2 管理菜单选项**执行 SQL 语句**并输入以下 SQL 查询：

```
select current server from sysibm.sysdummy1
```
- e. 创建存储器组并记录名称，例如 OBSVRSG。
- f. 如果要使用新数据库，那么请创建一个新数据库，例如将其命名为 OBSVRDB。根据需要，可以重复使用现有的数据库和存储器组，例如使用业务流程编排器数据库 BPEDB。
- g. 确定要使用的模式限定符（\_SQLID）。
- h. 确定用于设置数据库的用户标识 *user\_ID*。这并不是在运行时用于访问此数据库的用户标识。
- i. 确保该用户标识具有下列用于访问该数据库和存储器组的权限：
  - 有权使用该存储器组。
  - 有权使用数据库 OBSVRDB。
  - 有权在数据库 OBSVRDB 中创建表空间。
  - 有权在数据库 OBSVRDB 中创建表。
- j. 如果要使用业务流程编排器用户定义函数（UDF）的 Java 实现，那么请确保用户标识还具有下列权限：
  - 有权对 SYSIBM.SYSJAROBJECTS 执行查询。
  - 有权对模式 SQLJ 执行下列存储过程：
    - INSTALL\_JAR
    - REMOVE\_JAR
    - REPLACE\_JAR

- DB2\_INSTALL\_JAR
  - DB2\_REMOVE\_JAR
  - DB2\_REPLACE\_JAR
  - 有权执行属于集合 DSNJAR 的软件包。
- k. 如果要使用业务流程编排器用户定义函数 (UDF) 的 Java 实现, 那么请准备 DB2 环境, 以运行 Java 用户定义函数和解释型 Java 例程。请执行下列操作:
- 1) 启用 DB2 提供的存储过程并定义 DB2 通用 JDBC 驱动程序使用的表
  - 2) 为解释型 Java 例程设置环境
- 记录此过程中创建的 WLM 应用程序环境的名称。
2. 登录到 USS。
  3. 创建表空间:
    - a. 切换到所使用数据库系统的业务流程编排器观察器数据库脚本的目录。根据 DB2 版本的不同, 输入下列其中一个命令:
 

```
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV7
cd install_root/dbscripts/ProcessChoreographer/DB2zOSV8
```
    - b. 编辑 createTablespace\_Observer.sql 脚本。将 @STOGRP@ 替换为存储器组名, 并将 @DBNAME@ 替换为数据库名称而非子系统名称。
 

**注:** SQL 文件以 ASCII 格式交付。根据您查看、编辑或运行此文件时所使用工具的不同, 可能需要将该文件转换为 EBCDIC 格式。要将该文件转换为 EBCDIC 格式, 请输入以下命令:

```
iconv -t IBM-1047 -f ISO8859-1 createTablespace_Observer.sql > createTablespace_Observer_EBCDIC.sql
```

要将该文件重新转换为 ASCII 格式, 请输入以下命令:

```
iconv -t ISO8859-1 -f IBM-1047 createTablespace_Observer_EBCDIC.sql > createTablespace_Observer_ASCII.sql
```
    - c. 确保已连接至数据库, 然后运行经过定制的 createTablespace\_Observer.sql 脚本版本。
  4. 启动用于设置事件收集器的工具, 如第 238 页的『setupEventCollector 工具』所述。
  5. 准备数据库: 当您看到:
    - 1) Prepare a database for the Event Collector and Observer
    - 2) Install the Event Collector application
    - 3) Remove the Event Collector application and related objects
    - 4) Change configuration settings of an installed Event Collector
    - 5) Drop the database schema of the Event Collector and Observer
    - 6) Administer Observer related user-defined functions
    - 0) Exit Menu
    - a. 选择选项 1 以便为事件收集器应用程序准备数据库。
    - b. 输入 7 或 8 以选择 DB2 z/OS 版的版本号。
    - c. 此工具允许您创建 SQL 文件, 接着, 可以将此文件交给数据库管理员运行, 而不是使用您的当前用户标识来运行此文件。当您看到:
 

```
Do you want to create an SQL file only (delay database preparation)?
y) yes
n) no
```

      - 如果您不想推迟运行 SQL, 那么请输入 n。
      - 如果您想推迟运行 SQL, 那么请输入 y。您将看到:



Even if you want to delay the configuration,  
your entered values can be checked within the database.  
Do you want to perform these checks?  
y) yes  
n) no

- 如果您希望在数据库中对输入的值进行检查, 那么请输入 y。
- 否则, 请输入 n。

根据所输入内容的不同, 您可能不会看到下列所有提示。请跳过您未看到的步骤。

d. 如果您看到:

Specify the database location name:  
(as returned by SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1):

请输入数据库的位置名。这是您在第 203 页的 1d 步骤中记录的值。

e. 如果您看到:

Specify the name of the database as known by the subsystem [*subsystem*]

请输入该数据库在 z/OS 主机上的子系统具有的名称。这是您在第 203 页的 1f 步骤中记录的值。

f. 如果您看到:

Specify the hostname of the z/OS DB2 database server: [*localhost*]

请输入数据库服务器的主机名。

g. 如果您看到:

Specify the port where the database subsystem is listening:

请输入数据库子系统使用的端口号。这是您在第 203 页的 1c 步骤中记录的值。

h. 如果您看到:

Specify userid to connect to the database '*database\_alias*' [*db2admin*] :

请输入用于连接至该数据库的用户标识。这是用户标识 *user\_ID*, 如第 203 页的 1h 步骤所述。

i. 如果您看到:

Specify the password for userid '*user\_ID*' :

请输入该用户标识的密码。

j. 如果您看到:

Trying to connect to database '*database\_alias*', using user '*user\_ID*'  
Connected to '*database\_alias*'  
Specify the database schema to be used. [*user\_ID*] :

请输入数据库对象使用的数据库模式, 或者按 Enter 键接受缺省值, 即用于连接至该数据库的用户标识。这是模式限定符 *\_SQLID*。

k. 如果您看到:

Note: The Java UDFs are more precise, but they require a jar file  
installed to the database.  
Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 如果要使用较为精确的基于 Java 的用户定义函数 (UDF)，那么请输入 1，这将要求在数据库中安装一个 JAR 文件。
- 如果要使用不太精确的基于 SQL 的 UDF，那么请输入 2。

l. 如果您看到:

Specify the DB2 storage group name to be used. [OBSVRSG] :

请输入第 203 页的 1e 步骤中使用的存储器组名，或者按 Enter 键接受缺省值。

m. 如果您看到:

Specify the WLM environment name where the UDF should run. [] :

请输入您在第 204 页的 1k 步骤中记录的 WLM 环境。

n. 在检查必需的表空间并将 JAR 文件装入数据库后，以下消息将指示操作成功:

The setup of the database completed successfully.

6. 通过使用管理控制台，创建指向该数据库的 XA 数据源。

## 结果

业务流程编排器观察器的数据库模式已准备完毕。

### 相关概念

第 222 页的『业务流程编排器观察器的用户定义函数』

借助业务流程编排器观察器，可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告，业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数 (UDF)。

### 相关任务

第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』

通过使用 setupEventCollector 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数 (UDF) 与基于 SQL 的 UDF 之间进行切换。

## 从远程系统为业务流程编排器观察器创建 DB2 z/OS 版数据库:

本节描述如何在 Linux、UNIX 或 Windows 系统上使用菜单驱动的交互式工具以及 createTablespace\_Observer.sql 脚本来创建 DB2 z/OS 版数据库。

## 开始之前

必须已在 Linux、UNIX 或 Windows 服务器上安装了 WebSphere Process Server。

## 过程

1. 在数据库所在的 z/OS 服务器上:
  - a. 登录到本机 z/OS 环境。
  - b. 如果安装了多个 DB2 系统，那么请确定要使用的子系统。
  - c. 记录 DB2 子系统所侦听的 IP 端口。
  - d. 创建存储器组并记录名称，例如 OBSVRSG。

- e. 如果要使用新数据库，那么请创建一个新数据库，例如将其命名为 OBSVRDB。根据需要，可以重复使用现有的数据库和存储器组，例如使用业务流程编排器数据库 BPEDB。
- f. 确定要使用的模式限定符 (`_SQLID`)。
- g. 确定用于设置数据库的用户标识 `user_ID`。这并不是在运行时用于访问此数据库的用户标识。
- h. 确保该用户标识具有下列用于访问该数据库和存储器组的权限：
  - 有权使用该存储器组。
  - 有权使用数据库 OBSVRDB。
  - 有权在数据库 OBSVRDB 中创建表空间。
  - 有权在数据库 OBSVRDB 中创建表。
- i. 如果要使用业务流程编排器用户定义函数 (UDF) 的 Java 实现，那么请确保用户标识还具有下列权限：
  - 有权对 SYSIBM.SYSJAROBJECTS 执行查询。
  - 有权对模式 SQLJ 执行下列存储过程：
    - INSTALL\_JAR
    - REMOVE\_JAR
    - REPLACE\_JAR
    - DB2\_INSTALL\_JAR
    - DB2\_REMOVE\_JAR
    - DB2\_REPLACE\_JAR
  - 有权执行属于集合 DSNJAR 的软件包。
- j. 如果要使用业务流程编排器用户定义函数 (UDF) 的 Java 实现，那么请准备 DB2 环境，以运行 Java 用户定义函数和解释型 Java 例程。请执行下列操作：
  - 1) 启用 DB2 提供的存储过程并定义 DB2 通用 JDBC 驱动程序使用的表
  - 2) 为解释型 Java 例程设置环境

记录此过程中创建的 WLM 应用程序环境的名称。

2. 在 WebSphere Process Server 所在的服务器上：

- a. 安装合适的 DB2 客户机。

**注：**如果您计划使用本机 DB2 客户机通过 2 类 JDBC 连接来连接至远程数据库，那么请确保已安装 DB2 Connect Gateway。DB2 Connect Gateway 包括在 DB2 UDB ESE 软件包中，但也可以单独安装。

- b. 如果正在使用本机 DB2 客户机，那么请对远程数据库进行编目并验证能否连接至该数据库。在 DB2 命令行窗口中输入下列命令：

```
catalog tcpip node zosnode remote host_name server IP_port ostype mvs
catalog database location as database_alias at node zosnode
authentication dcs
catalog dcs database database_alias parms ',,INTERRUPT_ENABLED'
```

其中：

`zosnode`

是远程 z/OS 节点的本地别名。

*host\_name*

是远程 z/OS 机器的 TCP/IP 地址或别名。

*IP\_port*

是 DB2 子系统所侦听的端口号。

*database\_alias*

是用于访问远程数据库的本地别名。

*location*

是远程 DB2 位置名。要了解位置名，请登录到 TSO，然后使用其中一个可用的查询工具在所选子系统上输入以下 SQL 查询。

```
select current server from sysibm.sysdummy1
```

要验证能否连接至远程系统，请输入下列命令：

```
db2 connect to database_alias user userid using password
```

- c. 切换到所使用数据库系统的业务流程编排器观察器数据库脚本的目录：
  - 在 Linux 和 UNIX 平台上，根据 DB2 版本的不同，输入下列其中一个命令：

```
cd install_root/dbscripts/ProcessChoreographer/DB2z0SV7
```

```
cd install_root/dbscripts/ProcessChoreographer/DB2z0SV8
```
  - 在 Windows 平台上，根据 DB2 版本的不同，输入下列其中一个命令：

```
cd install_root\dbscripts\ProcessChoreographer\DB2z0SV7
```

```
cd install_root\dbscripts\ProcessChoreographer\DB2z0SV8
```
- d. 编辑 createTablespace\_Observer.sql 脚本。将 @STOGRP@ 替换为存储器组名，并将 @DBNAME@ 替换为数据库名称而非子系统名称。
- e. 运行经过定制的 createTablespace\_Observer.sql 脚本版本。如果要删除表空间，请使用 dropTablespace\_Observer.sql 脚本。
- f. 切换到配置脚本所在的业务流程编排器子目录。

在 Linux 和 UNIX 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\config
```

- g. 启动用于设置事件收集器的工具，如第 238 页的『setupEventCollector 工具』所述。
- h. 选择选项 1 以便为事件收集器应用程序准备数据库。
- i. 输入 7 或 8 以选择 DB2 z/OS 版的版本号。
- j. 此工具允许您创建 SQL 文件，接着，可以将此文件交给数据库管理员运行，而不是使用您的当前用户标识来运行此文件。当您看到：

```
Do you want to create an SQL file only (delay database preparation)?
```

```
y) yes
```

```
n) no
```

  - 如果您不想推迟运行 SQL，那么请输入 n。
  - 如果您想推迟运行 SQL，那么请输入 y。您将看到：

Even if you want to delay the configuration,  
your entered values can be checked within the database.  
Do you want to perform these checks?  
y) yes  
n) no

- 如果您希望在数据库中对输入的值进行检查, 那么请输入 y。
- 否则, 请输入 n。

根据所输入内容的不同, 您可能不会看到下列所有提示。请跳过您未看到的步骤。

k. 如果您看到:

Specify the JDBC driver type to be used:

- 2) Connect using type 2 (using a native database client)
- 4) Connect using type 4 (directly via JDBC)

请指定 JDBC 驱动程序类型:

- 如果正在使用本机数据库客户机, 请输入 2。
- 否则, 请输入 4 以选择 4 类 JDBC 驱动程序。

l. 如果您看到:

Specify the name of database in local catalog: [BPEDB]

请输入本地 DB2 客户机上编目该数据库时使用的数据库名称, 这是您在第 207 页的 2b 步骤中使用的 *database\_alias* 值。

m. 如果您看到:

Specify the location name/connection target: []

请输入所要连接的子系统的位置名。

**注:** 要确定位置名, 请以 SQL 处理者身份登录并执行以下 SQL 语句:

```
SELECT CURRENT SERVER FROM SYSIBM.SYSDUMMY1
```

n. 如果您看到:

Specify the name of the database as known by the subsystem: [OBSVRDB]

请输入用于在 z/OS 主机上的子系统中标识该数据库的名称。

o. 如果您看到:

Specify the hostname of the database server: [localhost]  
Specify the port where the database server is listening: [446]

请输入 z/OS 数据库服务器所使用的主机名和端口号。

p. 如果您看到:

Specify the directory of your JDBC driver: []

请输入 DB2 JDBC 驱动程序的 JAR 文件 db2jcc.jar 和 db2jcc\_license\_cisuz.jar 所在的目录。

q. 如果您看到:

Specify userid to connect to the database 'database\_name' [db2admin] :  
Specify the password for userid 'user\_ID' :

请输入用于连接至数据库的用户标识和密码。这是用户标识 `user_ID`，如第 207 页的 1g 步骤所述。

r. 如果您看到:

Specify the database schema to be used. [user\_ID] :

请输入数据库对象所使用的数据库模式的名称。

s. 如果您看到:

Note: The Java UDFs are more precise, but they require a jar file installed to the database.  
Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 如果要使用较为精确的基于 Java 的用户定义函数 (UDF)，那么请输入 1，这将要求在数据库中安装一个 JAR 文件。
- 如果要使用不太精确的基于 SQL 的 UDF，那么请输入 2。

t. 如果您看到:

Specify the DB2 storage group name to be used. [OBSVRSG] :

请输入第 206 页的 1d 步骤中使用的存储器组名。

u. 如果您看到:

Specify the WLM environment name where the UDF should run. [] :

请输入您在第 207 页的 1j 步骤中记录的 WLM 环境。在检查必需的表空间并将 JAR 文件装入数据库后，以下消息将指示操作成功:

The setup of the database completed successfully.

3. 通过使用管理控制台，创建指向该数据库的 XA 数据源。

## 结果

业务流程编排器观察器的数据库模式已准备完毕。

### 相关概念

第 222 页的『业务流程编排器观察器的用户定义函数』

借助业务流程编排器观察器，可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告，业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数 (UDF)。

### 相关任务

第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』

通过使用 `setupEventCollector` 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数 (UDF) 与基于 SQL 的 UDF 之间进行切换。

## 为业务流程编排器观察器准备 Derby 数据库

可以使用脚本或交互式工具来准备数据库。

### 使用 SQL 脚本为业务流程编排器观察器准备 Derby 数据库:

本节描述如何使用 `createSchema_Observer.sql` 脚本来准备 Derby 数据库。

## 关于此任务

您必须为业务流程编排器观察器数据库创建模式。可以在现有数据库中创建此模式，也可以让脚本文件创建新数据库。

## 过程

1. 切换到数据库的配置脚本所在的业务流程编排器子目录。
  - 在 Linux、UNIX 和 i5/OS 平台上，切换到 `install_root/dbscripts/ProcessChoreographer/Derby` 目录。
  - 在 Windows 平台上，切换到 `install_root\dbscripts\ProcessChoreographer\Derby` 目录。
2. 请执行下列操作：
  - a. 在 Derby 网络服务器环境中，将 `*Observer.sql` 脚本复制到网络服务器。并且，将 JAR 文件 `bpcodbutil.jar` 从 `install_root` 目录的 `lib` 子目录复制到数据库服务器上的同一目录中。
  - b. 在文本编辑器中，阅读 `createSchema_Observer.sql` 脚本文件开头的指示信息。如果要创建新数据库，那么请在数据库名称后面追加 `;create=true`。例如，如果数据库名称是 `OBSVRDB`，那么请将参数 `-Dij.database=OBSVRDB` 替换为 `-Dij.database=OBSVRDB;create=true`。

**注：**在 Windows 平台上，请不要使用记事本编辑器，此编辑器无法以可阅读的格式显示脚本文件。
  - c. 如果使用嵌入式 Derby 驱动程序连接至现有数据库，那么请停止服务器以及任何其他使用该数据库的应用程序。
  - d. 创建模式。在创建数据库的目录中，按照脚本文件 `createSchema_Observer.sql` 开头的指示信息运行此文件。
  - e. 如果发生了错误，那么可以运行脚本文件 `dropSchema_Observer.sql` 来删除此模式。
3. 使用管理控制台来创建指向该数据库的 XA 数据源，然后测试连接。

## 结果

业务流程编排器观察器的数据库模式已准备完毕。

### 使用 **setupEventCollector** 工具为业务流程编排器观察器准备 **Derby** 数据库：

本节描述如何在任何受支持平台上使用菜单驱动的交互工具 `setupEventcollector` 来准备 Derby 数据库。

## 过程

1. 切换到配置脚本所在的业务流程编排器子目录。

在 Linux、UNIX 和 i5/OS 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\config
```



2. 如果使用嵌入式 Derby 驱动程序连接至现有数据库，那么请停止服务器以及任何其他使用该数据库的应用程序。进行规划，以便在启动此工具时指定 `-conntype none`。
3. 启动用于设置事件收集器的工具，如第 238 页的『`setupEventCollector` 工具』所述。
4. 当您看到:
  - 1) Prepare a database for the Event Collector and Observer
  - 2) Install the Event Collector application
  - 3) Remove the Event Collector application and related objects
  - 4) Change configuration settings of an installed Event Collector
  - 5) Drop the database schema of the Event Collector and Observer
  - 6) Administer Observer related user-defined functions
  
  - 0) Exit Menu

选择选项 1 以便为事件收集器应用程序准备数据库。将显示以下菜单:

```
Prepare a database for the WebSphere Business Process Choreographer
Event Collector and Observer
```

Select the type of your database provider:

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) DB2 V7 on z/OS
- 8) DB2 V8 on z/OS
- o) Oracle
  
- 0) Exit Menu

5. 输入 `c` 以选择 Derby。
6. 此工具允许您创建 SQL 文件，接着，可以将此文件交给数据库管理员运行，而不是使用您的当前用户标识来运行此文件。当您看到:

```
Do you want to create an SQL file only (delay database preparation)?
```

- y) yes
- n) no

- 如果您不想推迟运行 SQL，那么请输入 `n`。
- 如果您想推迟运行 SQL，那么请输入 `y`。您将看到:

```
Even if you want to delay the configuration,
your entered values can be checked within the database.
```

```
Do you want to perform these checks?
```

- y) yes
- n) no

- 如果您希望在数据库中对输入的值进行检查，那么请输入 `y`。
- 否则，请输入 `n`。

根据所输入内容的不同，您可能不会看到下列所有提示。请跳过您未看到的步骤。

7. 如果您看到:

```
Specify the JDBC driver type to be used:
```

- 1) Connect using the embedded JDBC driver
- 2) Connect using the network JDBC driver

```
Your selection: [1]
```

- 要使用嵌入式 JDBC 驱动程序进行连接，请输入 `1`。

**要点:** 使用此驱动程序配置数据库时, 请确保没有任何其他应用程序 (包括 WebSphere Process Server) 连接至该数据库。

- 要使用网络 JDBC 驱动程序, 请输入 2。

8. 当您看到: Specify the name of your database [*database\_name*]

请输入该数据库的标准路径。

**注:** 缺省值 ... \BPEDB 就是业务流程编排器所使用的数据库。为了提高性能, 请使用另一个数据库。

9. 如果您看到:

Specify the database schema to be used. [APP] :

输入数据库对象所使用的数据库模式的名称。如果输入了空格字符或者未填写此字段, 那么将使用缺省模式 APP。

10. 如果您看到:

Specify the hostname of the database server: [localhost]  
Specify the port where the database server is listening: [1527]

请输入 Derby 网络服务器的主机名和端口号。

11. 如果您看到:

Specify the directory of your JDBC driver: [B:\w\p\derby\lib]

- 对于嵌入式 JDBC 驱动程序, 请输入 derby.jar 文件所在的目录。
- 对于网络 JDBC 驱动程序, 请输入 derbyclient.jar 所在的目录。

12. 如果您看到:

Specify userid to connect to the database *database\_name*: []

- 如果服务器要求进行认证, 那么请输入有权连接到 Derby 网络服务器的用户标识。
- 如果不输入值, 那么将使用用户标识 dummy。这是因为, Derby JDBC 驱动程序始终需要用户标识才能连接到网络服务器。

13. 如果您看到:

The application server must be stopped to update a Derby /  
Cloudscape database.  
This must be done outside wsadmin using 'stopServer *server\_name*'.  
After the server is stopped, come back to this prompt and enter  
'c' to continue.  
Please stop the server '*server\_name*' now.  
Press 'c' to continue, 'a' to abort:

- a. 在 wsadmin 外部, 使用以下命令来停止该服务器:

```
stopServer server_name
```

- b. 如果已停止该服务器, 请按 c 继续。否则, 按 a 返回到第 212 页的 4 步骤中显示的主菜单。

14. 如果您看到:

Specify the database schema to be used. [APP] :

输入数据库对象使用的模式的名称, 或者按 Enter 键以使用缺省值。

15. 确保您看到以下消息, 此消息确认数据库已成功地准备完毕:

The setup of the database completed successfully.

16. 如果您看到:

```
Restart the server now using 'startServer server_name'.
After the server is up again, come back to this prompt and enter
'c' to continue.
Press 'c' to continue, 'a' to abort:
```

a. 使用以下命令来启动该服务器:

```
startServer server_name
```

b. 请等待服务器启动完毕, 然后返回到此提示, 按 c 键继续。否则, 按 a 返回到第 212 页的 4 步骤中显示的主菜单。

以下消息指示操作成功:

```
WASX7074I: 已将 SOAP 连接器重新连接到主机 localhost.
```

17. 使用管理控制台来创建指向该数据库的 XA 数据源, 然后测试连接。

## 结果

业务流程编排器观察器的数据库模式已准备完毕。

## 为业务流程编排器观察器准备 Oracle 数据库

可以使用脚本或交互式工具来准备数据库。

### 使用 SQL 脚本为业务流程编排器观察器准备 Oracle 数据库:

本节描述如何在 Linux、UNIX 和 Windows 平台上使用 createTablespace\_Observer.sql 和 createSchema\_Observer.sql 脚本来准备 Oracle 数据库。

### 关于此任务

该数据库必须已存在。您可以使用现有数据库, 也可以根据数据库文档创建新数据库。

### 过程

1. 切换到数据库的配置脚本所在的业务流程编排器子目录。
  - 在 Linux、UNIX 和 i5/OS 平台上, 切换到 `install_root/dbscripts/ProcessChoreographer/Oracle` 目录。
  - 在 Windows 平台上, 切换到 `install_root\dbscripts\ProcessChoreographer\Oracle` 目录。
2. 将所有 \*Observer.sql 脚本文件复制至数据库服务器。
3. 如果要使用基于 Java 的业务流程编排器用户定义函数:
  - a. 将 JAR 文件 `bpcodbutil.jar` 从 `install_root` 目录的 `lib` 子目录复制到 SQL 脚本文件所在的目录。
  - b. 安装包含业务流程编排器观察器 UDF 的 JAR 文件。
    - 1) 作为具有 Oracle 管理权的用户登录到数据库服务器, 然后切换到 JAR 文件 `bpcodbutil.jar` 所在的目录:
      - 如果数据库在应用程序服务器所在的服务器上, 那么请切换到 `install_root` 目录的 `lib` 子目录。
      - 如果数据库不在应用程序服务器所在的机器上, 那么请切换到 JAR 文件 `bpcodbutil.jar` 被复制到的目录。

- 2) 通过输入以下命令运行 Oracle 的 loadjava 实用程序，以安装 JAR 文件 bpcodbutil.jar:

```
loadjava -user user/password@database
 -schema schema_name
 -resolve bpcodbutil.jar
```

其中:

*user*、*password* 和 *database* 是有效的用户标识、密码和数据库名称值。

*schema\_name* 是用于存储类的模式的名称。此模式必须是 Observer 表所使用的模式。

- 3) 如果发生问题，那么可以使用以下命令删除该 JAR 文件:

```
dropjava -user user/password@database
 -schema schema_name bpcodbutil.jar
```

#### 4. 创建表空间

- a. 按照 createTablespace\_Observer.sql 脚本文件开头的指示信息，编辑此文件的副本。
- b. 按照 createTablespace\_Observer.sql 脚本文件开头的指示信息，运行此文件的副本。
- c. 确保脚本输出未包含任何错误。如果发生了错误，那么可以使用 dropTablespace\_Observer.sql 脚本文件删除该表空间。

#### 5. 创建模式（表、索引和视图）。

- a. 按照 createSchema\_Observer.sql 脚本文件开头的指示信息，编辑并运行此文件的副本。
- b. 确保脚本输出未包含任何错误。如果要删除该模式，请使用 dropSchema\_Observer.sql 脚本文件。

#### 6. 使用管理控制台来创建指向该数据库的 XA 数据源，然后测试连接。

### 结果

业务流程编排器观察器的数据库模式已准备完毕。

#### 相关概念

第 222 页的『业务流程编排器观察器的用户定义函数』

借助业务流程编排器观察器，可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告，业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数（UDF）。

#### 相关任务

第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』

通过使用 setupEventCollector 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数（UDF）与基于 SQL 的 UDF 之间进行切换。

### 使用 **setupEventCollector** 工具为业务流程编排器观察器准备 **Oracle** 数据库:

本节描述如何在 Linux、UNIX 和 Windows 平台上使用菜单驱动的交互式工具和 createTablespace\_Observer.sql 脚本来准备 Oracle 数据库。

## 关于此任务

该数据库必须已存在。您可以使用现有数据库，也可以根据数据库文档创建新数据库。

**注：**要从 i5/OS 平台中创建远程 Oracle 数据库，请执行第 214 页的『使用 SQL 脚本为业务流程编排器观察器准备 Oracle 数据库』。

## 过程

1. 在 Linux 和 UNIX 平台上：将 \$ORACLE\_HOME/bin 添加至 PATH 变量。
2. 创建表空间：
  - a. 切换到数据库的配置脚本所在的业务流程编排器子目录。
    - 在 Linux 和 UNIX 平台上，切换到 `install_root/dbscripts/ProcessChoreographer/Oracle` 目录。
    - 在 Windows 平台上，切换到 `install_root\dbscripts\ProcessChoreographer\Oracle` 目录。
  - b. 按照 `createTablespace_Observer.sql` 脚本文件开头的指示信息，编辑此文件。
  - c. 按照表空间创建脚本文件开头的指示信息，运行此文件。
  - d. 确保脚本输出未包含任何错误。如果发生了错误，那么可以使用 `dropTablespace_Observer.sql` 脚本文件删除该表空间。
3. 切换到配置脚本所在的业务流程编排器目录。

在 Linux 和 UNIX 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\config
```

4. 启动用于设置事件收集器的工具，如第 238 页的『setupEventCollector 工具』所述。
5. 准备数据库：当您看到：
  - 1) Prepare a database for the Event Collector and Observer
  - 2) Install the Event Collector application
  - 3) Remove the Event Collector application and related objects
  - 4) Change configuration settings of an installed Event Collector
  - 5) Drop the database schema of the Event Collector and Observer
  - 6) Administer Observer related user-defined functions
  
  - 0) Exit Menu

请执行下列操作：

- a. 请选择选项 1，以便为事件收集器和观察器应用程序准备数据库。将显示以下菜单：

```
Prepare a database for the WebSphere Business Process Choreographer
Event Collector and Observer
```

```
Select the type of your database provider:
```

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) DB2 V7 on z/OS

- 8) DB2 V8 on z/OS
- o) Oracle

0) Exit Menu

- b. 输入 o 以选择 Oracle。
- c. 此工具允许您创建 SQL 文件，接着，可以将此文件交给数据库管理员运行，而不是使用您的当前用户标识来运行此文件。当您看到：

Do you want to create an SQL file only (delay database preparation)?

- y) yes
- n) no

- 如果您不想推迟运行 SQL，那么请输入 n。
- 如果您想推迟运行 SQL，那么请输入 y。您将看到：

Even if you want to delay the configuration,  
your entered values can be checked within the database.  
Do you want to perform these checks?

- y) yes
- n) no

- 如果您希望在数据库中对输入的值进行检查，那么请输入 y。
- 否则，请输入 n。

根据所输入内容的不同，您可能不会看到下列所有提示。请跳过您未看到的步骤。

- d. 如果您看到：

Specify the database to be used.

Note: Database must already exist.

Specify the name of your database [BPEDB] :

请输入数据库的 SID 名称。

- e. 如果您看到：

Specify the hostname where the oracle database resides: [localhost]

请输入数据库服务器的主机名或 IP 地址。

- f. 如果您看到：

Specify the port where the oracle listener is listening: [1521]

请输入 Oracle 侦听器的端口号。

- g. 如果您看到：

Specify userid to connect to the database '*database\_name*' [system] :

请输入用于连接至该数据库的用户标识。缺省值是 system。

- h. 如果您看到：

Specify the password for userid '*user\_ID*' :

请输入该用户标识的密码。

- i. 如果您看到：

Choose the implementation of the Observer user-defined functions.

Note: The Java UDFs are more precise, but they require a jar file installed to the database.

Visit the Observer documentation for details.

- 1) Java
- 2) SQL

0) Exit Menu

- 如果要使用较为精确的基于 Java 的用户定义函数 (UDF)，那么请输入 1，这将要求在数据库中安装一个 JAR 文件。
- 如果要使用不太精确的基于 SQL 的 UDF，那么请输入 2。

您将看到类似于以下的内容：

```
Trying to connect to database 'database_name', using user 'user_ID'
Connected to 'database_name'
Checking for required tablespace(s) ['OBSVRTS', 'OBSVRL0B', 'OBSVRIDX']
All required tablespaces were found.
Loading the jar file 'install_root\lib\bpcodbutil.jar' into the database.
The jar file 'install_root\lib\bpcodbutil.jar' was successfully installed.
```

The setup of the database completed successfully.

6. 通过使用管理控制台，创建指向该数据库的 XA 数据源。

## 结果

业务流程编排器观察器的数据库模式已准备完毕。

### 相关概念

第 222 页的『业务流程编排器观察器的用户定义函数』

借助业务流程编排器观察器，可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告，业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数 (UDF)。

### 相关任务

『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』

通过使用 setupEventCollector 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数 (UDF) 与基于 SQL 的 UDF 之间进行切换。

## 在 Java 用户定义函数与 SQL 用户定义函数之间进行选择

通过使用 setupEventCollector 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数 (UDF) 与基于 SQL 的 UDF 之间进行切换。

### 相关概念

第 222 页的『业务流程编排器观察器的用户定义函数』

借助业务流程编排器观察器，可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告，业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数 (UDF)。

## 使用脚本在 Java 用户定义函数与 SQL 用户定义函数之间进行选择

本节描述如何使用脚本对业务流程编排器观察器数据库在基于 Java 的用户定义函数 (UDF) 与基于 SQL 的 UDF 之间进行切换。

### 关于此任务

您想要使用业务流程编排器观察器 UDF 的 Java 实现来代替 SQL 实现。



## 过程

1. 如果您打算使用业务流程编排器观察器用户定义函数 (UDF) 的 Java 实现, 那么请将 JAR 文件 `bpcodbutil.jar` 从 `install_root` 目录的 `lib` 子目录复制到数据库服务器上的同一目录。
2. 如果您打算使用业务流程编排器观察器 UDF 的 Java 实现, 那么请安装 JAR 文件 `bpcodbutil.jar`:

- a. 如果尚未连接到该数据库, 那么通过在 DB2 命令行处理器中输入以下命令来连接到该数据库:

```
db2 connect to databaseName
```

- b. 通过输入以下命令, 安装该 JAR 文件:

```
db2 call sqlj.install_jar('file:pathURL','schema.BPCODBUTIL')
```

其中, `pathURL` 是该 JAR 文件的标准 URL, `schema` 是业务流程编排器数据库的模式名。例如:

- 在 Linux 和 UNIX 平台上, 如果该 JAR 文件在 `/tmp` 目录中, 那么必须输入以下命令:

```
db2 call sqlj.install_jar('file:/tmp/bpcodbutil.jar',
 'schema.BPCODBUTIL')
```

- 在 Windows 平台上, 如果该 JAR 文件在 `c:\tmp` 目录中, 那么必须输入以下命令:

```
db2 call sqlj.install_jar('file:c:/tmp/bpcodbutil.jar',
 'schema.BPCODBUTIL')
```

**注:** 如果要删除该 JAR 文件, 请使用以下命令:

```
db2 call sqlj.remove_jar('schema.BPCODBUTIL')
```

3. 删除 UDF 的 SQL 实现。按照 `dropFunctions_Observer.sql` 脚本文件开头的指示信息, 编辑此文件。例如, 对于 DB2, 请在 DB2 命令行处理器中输入以下命令:

```
db2 -tf dropFunctions_Observer.sql
```

4. 创建 UDF 的 Java 实现。按照 `createFunctionsJava_Observer.sql` 脚本文件开头的指示信息, 编辑此文件。例如, 对于 DB2, 请在 DB2 命令行处理器中输入以下命令:

```
db2 -tf createFunctionsJava_Observer.sql
```

## 结果

所使用的 UDF 实现已进行切换。

## 使用 `setupEventCollector` 工具在 Java 用户定义函数与 SQL 用户定义函数之间进行选择

本节描述如何使用菜单驱动的交互式工具对业务流程编排器观察器数据库在基于 Java 的用户定义函数 (UDF) 与基于 SQL 的 UDF 之间进行切换。

## 关于此任务

对于 Derby 数据库而言, setupEventCollector 始终使用基于 Java 的 UDF。对于其他数据库类型而言, 在缺省情况下, setupEventCollector 将使用基于 Java 的 UDF, 但您可以使用此工具切换到基于 SQL 的 UDF。如果您再次改变主意, 那么可以使用此工具切换回到基于 Java 的 UDF。

## 过程

1. 启动用于设置事件收集器的工具, 如第 238 页的『setupEventCollector 工具』所述。您将看到以下菜单:

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions
  
- 0) Exit Menu

2. 请选择选项 6 以管理与观察器相关的用户定义函数。 您将看到以下菜单:

- c) Derby
- d) DB2 Universal
- i) DB2 iSeries
- 7) DB2 V7 on z/OS
- 8) DB2 V8 on z/OS
- o) Oracle

3. 如果正在使用 DB2 Linux 版、UNIX 版或 Windows 版或者 DB2 z/OS 版, 那么请选择数据库版本的相应选项: d、7 或 8

- a. 当您看到以下菜单时:

Specify which type should be used to connect to the Database:

- 2) Connect using type 2 (using a native DB2 client)
- 4) Connect using type 4 (directly via JDBC)

请选择下列其中一个选项:

- 2 代表使用本机 DB2 客户机的 2 类 JDBC 连接。在这种情况下, 将提示您输入下列内容:

**数据库名称**

**数据库用户标识**

**密码**

**JDBC 驱动程序的目录**

- 4 代表直接进行连接的 4 类 JDBC 驱动程序。在这种情况下, 将提示您输入下列内容:

**数据库名称**

**数据库服务器主机名**

**数据库服务器端口号**

**JDBC 驱动程序的目录**

**数据库用户标识**

**密码**

4. 如果正在使用 Oracle, 那么请选择选项 o。

- a. 输入以下连接信息:
  - 数据库服务器主机名
  - 数据库服务器端口号
  - 数据库名称
  - 数据库用户标识
  - 密码

#### JDBC 驱动程序 的目录

- 5. 如果能够建立与该数据库的连接，那么您将看到用于管理观察器数据库 UDF 的菜单:
  - 6) Administer Observer related user-defined functions
    - 1) Activate Java based user-defined functions
    - 2) Activate SQL based user-defined functions
    - 3) Determine current state
    - 4) List, install or remove the jar file containing the java based functions

注：『激活』选项不适用于 Derby 数据库。

- a. 如果要激活基于 Java 的 UDF，请选择选项 1。

- 1) 当您看到:

Specify the database schema to be used:

请输入数据库模式的名称。

- 2) 当您看到:

WARNING: Switching the UDF implementation type may break any running Observer applications. Continue anyway?

- y) yes
- n) no

Your selection:

请输入 y 继续，或者输入 n 停止。

- 3) 如果继续的话，您将看到类似于以下的内容:

Removing the user-defined functions ...

The jar file with jar\_id 'DB2INST1.BPCODBUTIL' is updated with the current version.  
Loading the jar file 'B:\w\p\lib\bpcodbutil.jar' into the database.  
The jar file 'BPCODBUTIL' was successfully installed.

Creating the Java based user-defined functions ...

- 4) 以下消息指示操作成功:

The setup of the database completed successfully.

- b. 如果要激活基于 SQL 的 UDF，请选择选项 2。

- 1) 当您看到:

Specify the database schema to be used:

请输入数据库模式的名称。

- 2) 当您看到:

```
WARNING: Switching the UDF implementation type may break any
running Observer applications. Continue anyway?
y) yes
n) no
Your selection:
```

请输入 y 继续, 或者输入 n 停止。

3) 如果您看到:

```
Removing the user-defined functions ...

Creating the SQL based user-defined functions ...

Do you also want to remove the jar file from the database?
y) yes
n) no
Your selection:
```

请输入 y 从该数据库中除去 JAR 文件, 或者输入 n 不除去该文件。

4) 以下消息指示操作成功:

```
The setup of the database completed successfully.
```

- c. 可选: 要确定所选 UDF 实现是基于 Java 还是基于 SQL, 并且还要在选择 Java 的情况下验证是否已安装 JAR 文件, 那么请选择选项 3。例如, 如果已选中 Java 实现, 那么您将接收到类似于以下的消息:

```
The active UDF implementation is Java.
Tested functionality of the UDF, is working
```

- d. 可选: 要安装或除去基于 Java 的 UDF 所需的 JAR 文件, 或者要列示数据库中安装的所有 JAR 文件, 请选择选项 4。然后, 当您看到以下菜单时:

```
List, install or remove jar files containing the java based functions
```

- ```
1) Install the jar file containing the Observer functions
   into the database
2) Remove the jar file containing the Observer functions
   from the database
3) List installed jar files

0) Exit Menu
```

- 选择选项 1 以安装该 JAR 文件。
- 选择选项 2 以除去该 JAR 文件。
- 选择选项 3 以列示数据库中安装的 JAR 文件。
- 选择选项 0 以退出菜单。

- e. 重复地选择选项 0 以返回到第 220 页的 1 步骤中显示的菜单。

结果

业务流程编排器观察器数据库将使用所选 UDF。

业务流程编排器观察器的用户定义函数

借助业务流程编排器观察器, 可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告, 业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数 (UDF)。

可以安装下列任何一种实现的 UDF:

SQL 实现

对于使用由数据库系统提供的内置时间函数以普通 SQL 实现的 UDF，请使用 SQL 实现。

由于 SQL 实现只要求运行所提供的 SQL 脚本，因此安装 SQL 实现比安装 Java 实现容易。对于这些脚本而言，进行安装时所需的管理权较少。此外，SQL 实现的性能高于 Java 实现。但是，由于内置时间函数有一些限制，因此 SQL 实现的 UDF 可能不够精确，无法满足您的需要。例如，在 DB2 中，内置时间函数假定每个月都是 30 天，这可能会导致结果不正确。

SQL 实现不适用于 Derby 数据库。

Java 实现

对于使用 Java 语言实现的 UDF，请使用 Java 实现。

要安装 Java 实现，请使用数据库系统提供的机制。Java 实现的 UDF 能够生成精确的报告。但是，与安装 SQL 实现相比，安装 Java 实现时需要执行的步骤较多，并且要求对数据库拥有更多的管理权。例如，对于 DB2 z/OS 版数据库而言，必须设置工作负载（WLM）环境才能运行 UDF。

根据所选数据库设置方式的不同，缺省实现也会有所变化：

- 如果将数据库设置为使用 SQL 脚本，或者使用“第一次访问时创建表”功能，那么缺省情况下将安装 SQL 实现。
- 如果将数据库设置为使用 setupEventCollector 工具或者使用“创建概要文件”向导中的业务流程编排器样本配置（仅在 Derby 数据库中提供），那么缺省情况下将安装 Java 实现。

进行初始设置后，可以更改 UDF 的实现，如第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』所述。

相关任务

第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』

通过使用 setupEventCollector 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数（UDF）与基于 SQL 的 UDF 之间进行切换。

第 197 页的『使用 SQL 脚本为业务流程编排器观察器准备 DB2 iSeries 版数据库』

本节描述如何在 i5/OS qshell 环境中使用 createSchema_Observer.sql 脚本来准备 DB2 iSeries 数据库。

第 198 页的『使用 setupEventCollector 工具为业务流程编排器观察器准备 DB2 iSeries 版数据库』

本节描述如何从 i5/OS Qshell 环境中使用菜单驱动的交互式工具来准备 DB2 iSeries 数据库。

第 200 页的『使用 setupEventCollector 工具从远程系统准备 DB2 iSeries 版数据库』

本节描述如何使用菜单驱动的交互式工具从远程 Linux、Windows 或 UNIX 系统为业务流程编排器观察器准备 DB2 iSeries 版数据库。

第 203 页的『在 USS 中为业务流程编排器观察器创建 DB2 z/OS 版数据库』

本节描述如何在 z/OS 机器上的 UNIX 系统服务（USS）中使用菜单驱动的交互工具以及 createTablespace_Observer.sql 脚本来创建 DB2 z/OS 版数据库。

第 206 页的『从远程系统为业务流程编排器观察器创建 DB2 z/OS 版数据库』

本节描述如何在 Linux、UNIX 或 Windows 系统上使用菜单驱动的交互式工具以及 createTablespace_Observer.sql 脚本来创建 DB2 z/OS 版数据库。

第 214 页的『使用 SQL 脚本为业务流程编排器观察器准备 Oracle 数据库』
本节描述如何在 Linux、UNIX 和 Windows 平台上使用 createTablespace_Observer.sql
和 createSchema_Observer.sql 脚本来准备 Oracle 数据库。

第 215 页的『使用 setupEventCollector 工具为业务流程编排器观察器准备 Oracle 数据库』

本节描述如何在 Linux、UNIX 和 Windows 平台上使用菜单驱动的交互式工具和
createTablespace_Observer.sql 脚本来准备 Oracle 数据库。

第 193 页的『使用 SQL 脚本为业务流程编排器观察器准备 DB2 通用数据库』
本节描述如何在 Linux、UNIX 和 Windows 平台上使用 createTablespace_Observer.sql
和 createSchema_Observer.sql 脚本来准备 DB2 通用数据库。

第 194 页的『使用 setupEventCollector 工具为业务流程编排器观察器准备 DB2 通用数据库』

本节描述如何在 Linux、UNIX 和 Windows 平台上使用菜单驱动的交互式工具和
createTablespace_Observer.sql 脚本来准备 DB2 通用数据库。

第 238 页的『setupEventCollector 工具』

使用 setupEventCollector 以交互方式配置或除去业务流程编排器事件收集器应用程序、
设置数据库以及管理数据库的用户定义函数。此工具使用 wsadmin 脚本。

配置业务流程编排器事件收集器应用程序

使用交互式工具或管理控制台来安装和配置事件收集器应用程序。

开始之前

必须在要安装事件收集器应用程序的部署目标上配置公共事件基础结构 (CEI)。

关于此任务

要配置业务流程编排器事件收集器，请执行下列其中一项操作：

使用 setupEventCollector 工具配置业务流程编排器事件收集器

本节描述如何使用菜单驱动的交互式工具在服务器或集群中安装和配置事件收集器应用程序。

过程

1. 切换到配置脚本所在的业务流程编排器子目录。

在 Linux、UNIX 和 i5/OS 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\config
```

2. 启动用于设置事件收集器的工具，如第 238 页的『setupEventCollector 工具』所述。
例如：

在 Linux 和 UNIX 平台上：

```
setupEventCollector.sh -server server1
```

在 i5/OS 平台上：

```
setupEventCollector -server server1
```

在 Windows 平台上:

```
setupEventCollector.bat -server server1
```

您将看到命令菜单:

Commands Menu

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined funtions

0) Exit Menu

3. 要安装业务流程编排器事件收集器应用程序, 请执行下列操作:

a. 选择选项 2。 将显示以下内容:

```
Create required objects and install the WebSphere Business Process  
Choreographer Event Collector application ...
```

b. 如果正在独立服务器上执行安装, 那么您将看到:

```
Working on node 'your_node_name', server 'your_server_name'.
```

c. 如果正在 Deployment Manager 上安装该应用程序, 那么必须从所有可用目标的列表中选择部署目标。 例如:

Select the deployment target to install to:

- 1) Cluster 'cluster1'
- 2) Node 'Node04', Server 'managed1'
- 3) Node 'Node04', Server 'managed2'

0) Exit Menu

d. 当此工具在部署目标上搜索现有的事件收集器安装版本时, 您将看到类似于以下的内容:

```
Searching for an already installed Event Collector on 'deployment_target'
```

e. 如果存在已安装的事件收集器应用程序实例, 那么您将看到:

```
Do you want to overwrite the existing application?
```

- o) Overwrite
- a) Abort

- 输入 o 以覆盖现有的事件收集器应用程序。您可以重新输入所有安装值, 事件收集器应用程序将被更新。

- 输入 a 退出, 而不安装事件收集器。

4. 当您看到:

```
Specify the JNDI name of the database where the WebSphere Business Process  
Choreographer Event Collector should store the collected events.
```

```
Enter '?' to get a list.
```

```
Your selection : [jdbc/BPEDB]
```

输入用于连接至该数据库的 JNDI 名称。也可以输入 ? 以获取所有已注册的数据源的列表。例如:

```
jdbc/BPEDB  
jdbc/DefaultEJBTimerDataSource  
jdbc/mediation/messageLog
```

5. 当您看到:

Specify the database schema to be used.
Enter a space character or leave empty to use the default schema of the
datasource. [] :

请输入事件收集器用于存储事件的数据库表的模式名。要将数据源定义的认证别名中指定的用户标识用作模式，请输入空格字符或者不填写此字段。

这将创建所有必需对象并安装企业应用程序。以下消息指示操作成功:

```
WebSphere Business Process Choreographer Event Collector  
installed successfully!
```

6. 如果未对服务器启用 CEI 日志记录功能，那么您将看到以下内容:

```
Checking if CEI event logging is enabled ...
```

```
Warning: The Business process container of server_name has CEI event  
logging disabled.  
To allow the Event Collector to work correctly, CEI event logging is required.  
Do you want to enable the CEI event logging on server_name? (y/n)
```

- 如果要想脚本对所指定服务器启用 CEI 日志记录功能，那么请输入 *y*。
- 如果不想让脚本对所指定服务器启用 CEI 日志记录功能，那么请输入 *n*。

注：当您开始使用业务流程编排器观察器时，CEI 日志记录功能必须处于启用状态，这一点十分重要。

7. 当显示以下提示时:

```
Do you want to save the changes? (y/n)
```

如果没有任何错误消息，请输入 *y* 以保存配置。如果发生了错误，请输入 *n* 以废弃更改并保留原始配置。检查名为 `setupEventCollector.log` 的日志文件，此文件在概要文件的 `logs` 目录中。

例如，在 Windows 上，如果概要文件名为 `myServer`，并且概要文件存储在 `install_root\profiles` 中，那么日志文件在 `install_root\profiles\myServer\logs` 目录中。

8. 输入 `0` 以退出菜单。

9. 激活更改:

- 如果启动此工具时指定了 `-conntype NONE` 选项，那么所作的更改将在服务器重新启动后生效。
- 如果启动此工具时未指定 `-conntype NONE` 选项，并且安装业务流程编排器事件收集器期间已对服务器启用 CEI 日志记录，那么请使用管理控制台来停止 `BPEContainer` 应用程序，然后将其重新启动。

结果

业务流程编排器事件收集器应用程序已安装并配置完毕。

使用管理控制台配置业务流程编排器事件收集器

本节描述如何使用管理控制台在给定的服务器或集群中安装业务流程编排器事件收集器的实例。

开始之前

已准备好业务流程编排器观察器数据库。

过程

1. 在管理控制台中，转到业务流程编排器事件收集器的配置页面：单击**服务器** → **集群** → **cluster_name** 或**服务器** → **应用程序服务器** → **server_name**，然后在**业务集成**中展开**业务流程编排器**并单击**业务流程编排器事件收集器**。
2. 要创建新配置，请执行下列操作：
 - a. 输入或选择下列字段的值：
 - 数据库实例名。
 - 模式名。
 - 启用或清除用于在第一次使用数据库时创建数据库表的选项。
 - 用于连接到数据库的用户名和密码。
 - 数据库服务器的主机名或 IP 地址。
 - 数据库服务器的端口号。
 - JDBC 提供程序。
 - 观察目标：
 - 受管业务流程编排器容器
 - 现有事件组名
 - 事件组名
 - b. 单击**应用**以部署该应用程序。
 - c. 如果发生问题，请检查 SystemOut.log 文件。否则，请将更改保存至主配置。
 - d. 启动该应用程序：单击**应用程序** → **企业应用程序**，选择应用程序 BPCECollector_scope (scope 指定部署目标)，然后单击**启动**。

结果

业务流程编排器事件收集器已配置完毕。

业务流程编排器事件收集器:

必须先配置事件收集器，然后才能使用业务流程编排器观察器。

要查看此管理控制台页面，请单击**服务器** → **集群** → **cluster_name** 或**服务器** → **应用程序服务器** → **server_name**，然后在**业务集成**中展开**业务流程编排器**，并单击**事件收集器配置**。

数据源:

在此部分中，可以配置业务流程编排器事件收集器的数据源。

编辑:

单击此项以编辑数据源。

测试连接:

测试与所选数据源的连接。

数据库实例:

数据库的名称。

属性	值
数据类型	字符串
缺省值	\${USER_INSTALL_ROOT}\databases\ BPOBEC00

模式名:

数据库模式的名称。

属性	值
数据类型	字符串
缺省值	无

创建表:

如果选中此选项，那么第一次访问该数据库时，将自动创建表。要使此选项生效，该数据库必须已存在，并且提供的用户名必须有权在该数据库中创建表和索引。

对于生产系统而言，建议您不要使用此选项。如果未选中此选项，那么将不会自动创建表，您必须通过运行脚本手动创建这些表。

属性	值
数据类型	复选框
缺省值	选中

用户名:

有权连接到数据库和修改数据的用户标识。

如果该用户标识有权在数据库中创建表和索引，那么在应用服务包或修订包后，将根据需要自动更新数据库模式。

属性	值
数据类型	字符串
缺省值	当前登录用户。

密码:

数据源用户标识的密码。

属性	值
数据类型	字符串
缺省值	无

服务器:

数据库服务器的地址。

请指定主机名或 IP 地址。

属性
数据类型
缺省值

值
字符串
无

端口:

数据库服务器使用的端口号。

属性
数据类型
缺省值

值
字符串
取决于所选 JDBC 提供程序。

提供程序:

JDBC 提供程序。

属性
数据类型
缺省值

值
下拉列表
DERBY_EMBEDDED

JMS 用户名:

属性
数据类型
缺省值

值
字符串
对业务流程编排器 JMS 用户指定的用户标识。

JMS 密码:

属性
数据类型
缺省值

值
字符串
对业务流程编排器 JMS 用户指定的密码。

观察目标:

在此部分，指定事件收集器的目标。

属性
数据类型
选项

值
单选按钮
• 受管业务流程编排器容器
• 现有事件组名
• 事件组名

受管业务流程编排器容器:

选择已配置的业务流程编排器容器。

属性
数据类型
内容

值
下拉列表
全部业务流程编排器配置

事件组概要文件列表:

选择组概要文件。

属性	值
数据类型	下拉列表

输入事件组概要文件列表:

输入事件组概要文件列表。

属性	值
数据类型	字符串
缺省值	无

配置业务流程编排器观察器应用程序

可以使用工具或管理控制台来配置业务流程编排器观察器应用程序。

开始之前

您已配置业务流程编排器事件收集器。

关于此任务

要配置业务流程编排器观察器，请执行下列其中一项操作：

使用 **setupObserver** 工具配置业务流程编排器观察器

本节描述如何使用菜单驱动的交互式工具来安装业务流程编排器观察器应用程序的实例，并描述如何对其进行配置以连接至特定事件收集器的数据源。

开始之前

已准备好业务流程编排器观察器数据库，并已使用管理控制台为其创建数据源。

过程

1. 切换到配置脚本所在的业务流程编排器子目录。

在 Linux、UNIX 和 i5/OS 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/config
```

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\config
```

2. 启动用于设置观察器的工具，如第 240 页的『**setupObserver** 工具』所述。您将看到以下菜单：

- 1) Install the Observer application
- 2) Remove the Observer application and related objects
- 3) Change configuration settings of an installed Observer
- 0) Exit Menu

3. 选择选项 1 以安装业务流程编排器观察器。将显示以下内容:

```
Create required objects and install the WebSphere Business Process  
Choreographer Observer application ...
```

4. 如果正在独立服务器上执行安装,那么您将看到:

```
Working on node 'your_node_name', server 'your_server_name'.
```

5. 如果正在 Deployment Manager 上安装该应用程序,那么必须从所有可用目标的列表中选择部署目标。例如:

```
Select the deployment target to install to:
```

```
1) Cluster 'cluster1'  
2) Node 'Node04', Server 'managed1'  
3) Node 'Node04', Server 'managed2'
```

```
0) Exit Menu
```

6. 当您看到:

```
Specify the JNDI name of the database containing the event tables.  
Enter '?' to get a list.  
Your selection : [jdbc/BPEDB]
```

输入用于连接至该数据库的 JNDI 名称。也可以输入 ? 以获取所有已注册的数据源的列表。例如:

```
jdbc/BPEDB  
jdbc/DefaultEJBTimerDataSource  
jdbc/mediation/messageLog
```

7. 当您看到:

```
Specify the database schema to be used.  
Enter a space character or leave empty to use the default schema of the  
datasource. [] :
```

请输入事件收集器用于存储事件的数据库表的模式名。要将数据源定义的认证别名中指定的用户标识用作模式,请输入空格字符或者不填写此字段。

这将创建所有必需对象并安装企业应用程序。以下消息指示操作成功:

```
WebSphere BPC Observer installed successfully!
```

8. 当系统提示 Do you want to save the changes? (y/n) 时,如果没有任何错误消息,请输入 y 以保存配置。如果发生了错误,请输入 n 以废弃更改并保留原始配置。检查名为 setupObserver.log 的日志文件,此文件在概要文件的 logs 目录中。例如,在 Windows 上,如果概要文件名为 myServer,并且概要文件存储在 *install_root*\profiles 中,那么日志文件在 *install_root*\profiles\myServer\logs 目录中。
9. 输入 0 以退出菜单。

结果

业务流程编排器观察器已安装并配置完毕。

使用管理控制台配置业务流程编排器观察器

本节描述如何使用管理控制台来安装业务流程编排器观察器应用程序的实例,并描述如何对其进行配置以连接至特定事件收集器的数据源。

开始之前

已配置业务流程编排器事件收集器。

过程

1. 在管理控制台中，转到“业务流程编排器观察器”配置页面：单击**服务器** → **集群** → **cluster_name** 或 **服务器** → **应用程序服务器** → **server_name**，然后在**业务集成**中展开**业务流程编排器**并单击**业务流程编排器观察器**。
2. 要创建新配置，请执行下列操作：
 - a. 单击**添加**。
 - b. 输入或选择下列字段的值：
 - 此实例的上下文根
 - 对此业务流程编排器事件收集器中的监视数据进行可视化有关这些配置参数的更多信息，请参阅第 124 页的『业务流程编排器观察器设置』。
 - c. 单击**应用**以部署该应用程序。
 - d. 如果发生问题，请检查 SystemOut.log 文件。否则，请将更改保存至主配置。
 - e. 启动该应用程序：单击**应用程序** → **企业应用程序**，选择应用程序 BPCObserver_scope (scope 指定部署目标)，然后单击**启动**。

结果

业务流程编排器观察器已配置完毕并可供使用。

下一步做什么？

通过重复本任务，可以在同一个部署目标或不同部署目标上配置多个业务流程编排器观察器实例。但是，每个实例都必须连接至各自的事件收集器数据源。

对业务流程编排器启用日志记录

本节描述如何对业务流程编排器启用公共事件基础结构（CEI）事件。

开始之前

要使用业务流程编排器观察器来监视业务流程事件，必须使业务流程能够发出公共事件基础结构（CEI）事件。请在对业务流程进行建模时指定此设置。为了正确地监视业务流程，必须至少发出『Process Started』事件。要获取可以使用业务流程编排器观察器监视的 CEI 事件的列表，请参阅业务流程事件。有关如何使业务流程能够发出 CEI 事件的信息，请参阅 WebSphere Integration Developer 信息中心。

关于此任务

如果已将业务流程编排器事件收集器安装在配置了业务流程编排器的部署目标上，那么安装应用程序时可以使用 setupEventCollector 工具来启用 CEI 日志记录。如果已使用管理控制台来安装业务流程编排器事件收集器，那么必须使用脚本或管理控制台来启用 CEI 日志记录。

要使用 Jython 脚本对业务流程编排器启用 CEI 日志记录，请执行第 233 页的『使用脚本对业务流程编排器启用日志记录』。

要使用管理控制台对业务流程编排器启用 CEI 日志记录，请执行第 261 页的『使用管理控制台启用公共基本事件和审计跟踪』。

结果

业务流程和活动的公共事件基础结构事件将被发出，并可以由业务流程编排器事件收集器接收。

使用脚本对业务流程编排器启用日志记录

本节描述如何使用 `setStateObserver.py` 脚本对业务流程编排器启用或禁用公共事件基础结构（CEI）或审计事件。

位置

`setStateObserver.py` 脚本在业务流程编排器的 `config` 目录中。

运行脚本

要运行 `setStateObserver` 脚本，请输入以下命令：

在 Linux 和 UNIX 平台上，输入以下命令：

```
install_root/bin/wsadmin.sh -lang jython  
-f install_root/ProcessChoreographer/config/setStateObserver.py
```

在 iOS 平台上，输入以下命令：

```
install_root/bin/wsadmin -lang jython  
-f install_root/ProcessChoreographer/config/setStateObserver.py
```

在 Windows 平台上，输入以下命令：

```
install_root\bin\wsadmin.bat -lang jython  
-f install_root\ProcessChoreographer\config\setStateObserver.py
```

参数

此脚本文件可以接收下列参数：

-conntype *NONE*

仅当应用程序服务器（对于独立应用程序服务器）或 Deployment Manager 未处于运行状态时，才应该指定此选项。

-node *nodeName*

其中，*nodeName* 是节点的名称。如果指定了集群，那么请不要指定此选项。

-server *serverName*

其中，*serverName* 是服务器的名称。如果指定了集群，那么请不要指定此选项。

-cluster *clusterName*

其中，*clusterName* 是集群的名称。在独立服务器环境中，请不要指定此选项。如果指定了节点和服务器，也不要指定此选项。

-profileName *profileName*

其中，*profileName* 是要使用的概要文件的名称。

-enable (CEI | AuditLog | CEI;AuditLog)

（可选）指定是否启用 CEI 日志记录和/或审计日志记录。

-disable (CEI | AuditLog | CEI;AuditLog)

(可选) 指定是否禁用 CEI 日志记录和/或审计日志记录。

-bfm

(可选) 指定对业务流程编排器中负责运行业务流程的业务流程管理器启用或禁用日志记录。

-htm

(可选) 指定对业务流程编排器中负责运行人员任务的人员任务管理器启用或禁用日志记录。

示例

要在 Linux 或 UNIX 平台上对 server1 上的业务流程事件启用 CEI 日志记录:

```
wsadmin.sh -lang jython -f setStateObserver.py -server server1 -enable CEI -bfm
```

注: 在 Windows 平台上, 请使用 wsadmin.bat; 在 i5/OS 平台上, 请使用 wsadmin。

更改业务流程编排器观察器的配置参数

对于启用验证功能以及提高性能而言, 调整业务流程编排器观察器和事件收集器应用程序的配置参数至关重要。

更改缺省值

缺省值更适合于生产系统, 而不那么适合于测试系统。如果您设置业务流程编排器的目的是进行开发或测试, 那么在验证配置是否可行之前, 最好更改下列配置参数:

- 将 BPCEventTransformerEventCount 的值更改为 0。
- 将 BPCEventTransformerToleranceTime 的值更改为 1。

进行这些更改可以确保, 即使发出事件的频率比生产系统低, 那些事件也会在一分钟内变为可用。

事件收集器的配置参数

通过调整数字参数, 可以影响事件变换器的触发频率以及提供给业务流程编排器观察器处理的事件的时效。

配置参数	数据类型/单位	缺省值	描述
ObserverSchemaName	字符串	未设置	此参数指定用作所有数据库对象的前缀的数据库模式。如果此参数为空, 那么缺省情况是, 将用于连接至数据库的用户标识用作前缀。请在管理控制台的数据源定义中设置此用户标识。如果对此参数指定了值, 那么数据源中指定的用户标识必须有足够的权限来访问此模式的数据库对象。

配置参数	数据类型/单位	缺省值	描述
BPCEventTransformer EventCount	整数/事件	500	<p>此参数指定事件数，达到此数目后，事件收集器将触发变换器，以便将收集到的事件变换为适合于观察器应用程序的格式。</p> <p>在进行开发、测试和实验时，缺省值可能过大，这将导致事件长时间不可观察。要使事件更快地可用，可以将此值设置为零。以后的每个事件都将触发变换器并在观察器中变为可视。如果将此值更改为零，那么当新事件生成时，将立即变换所有尚未变换的旧事件。对于生产系统，建议您不要使用零值。</p>
BPCEventTransformer MaxWaitTime	整数/分钟	10	<p>触发变换器前可以经过的最大时间长度 - 即使未达到 BPCEventTransformer EventCount 指定的事件数亦如此。</p>
BPCEventTransformer ToleranceTime	整数/分钟	10	<p>事件在观察器中可视前必须达到的最小时效，以分钟计。这使相关事件能够可靠地相关联。您应该避免使用零值，否则，有可能导致一个事件在其前趋事件未到达前被处理。</p> <p>在进行开发、测试和实验时，缺省值可能会过大，这将导致新事件在长达 10 分钟的时间内保持不可观察。如果将此值设置为 1，那么时效超过 1 分钟的所有已变换事件都将在观察器中可视。</p>
ObserverCreateTables	布尔值		<p>此参数指示当 EJB 第一次连接到数据库时，是否应该创建 Observer 模式。有效值为“true”和“false”。</p>

当事件收集器从公共事件基础结构（CEI）接收到与业务相关的事件时，会将该事件保存在数据库中。经过一段时间并接收到更多事件后，将启动变换器。变换器对存储的事件进行成批变换，然后使用可以用于生成报告的格式将这些事件写回到数据库。只有经过变换器处理的事件才可供观察器报告使用。

事件收集器每次接收到新事件时，如果下列条件中的任何一个条件成立，那么变换器将开始进行处理：

- 变换器上次启动后接收到的事件数大于 BPCEventTransformerEventCount 值。

- 变换器上次启动后经过的时间大于 BPCEventTransformerMaxWaitTime 值（以分钟计）。

如果减小这些值，那么事件将更快地可用于生成报告，但转换少量事件会增加成本。这要求您进行权衡，以确定是通过处理大量事件提高变换吞吐量，还是需要使事件尽快地在观察器数据库中可用。

变换器每次启动时，它都将处理所有比 BPCEventTransformerToleranceTime（以分钟计）旧的事件。它不会处理更新的事件，这是因为事件不一定按照它们的发生顺序发布。BPCEventTransformerToleranceTime 的缺省设置假定，接收任何事件并将其写入事件收集器表所耗用的时间不会超过 10 分钟。

更改事件收集器的配置参数

要更改事件收集器的参数，请执行下列操作：

1. 启动用于设置事件收集器的工具，如第 238 页的『setupEventCollector 工具』所述。您将看到以下菜单：
 - 1) Prepare a database for the Event Collector and Observer
 - 2) Install the Event Collector application
 - 3) Remove the Event Collector application and related objects
 - 4) Change configuration settings of an installed Event Collector
 - 5) Drop the database schema of the Event Collector and Observer
 - 6) Administer Observer related user-defined functions
 - 0) Exit Menu
2. 选择选项 4 以显示一组可以更改的参数：
 - 1) BPCEventTransformerEventCount
 - 2) BPCEventTransformerMaxWaitTime
 - 3) BPCEventTransformerToleranceTime
 - 4) ObserverCreateTables
 - 5) ObserverSchemaName
 - 0) Exit Menu
3. 选择要更改的参数的编号。将显示该参数的名称、描述、类型、单位和当前值。
4. 要更改指定的值，请输入新值，然后按 Enter 键。如果在未输入新值的情况下按 Enter 键，那么将返回参数列表。
5. 如果要更改另一个参数的值，请从步骤 3 重复。
6. 输入 0 以退出该列表。系统将询问您是否要保存更改。
7. 要保存所有更改，请输入 y，否则输入 n 以废弃所有更改。
8. 要激活更改，请重新启动 BPCECollector 应用程序。

观察器的配置参数

ReportAtSnapshotRange 参数值会对快照报告的性能产生巨大影响。

配置参数	数据类型/单位	缺省值	描述
ObserverSchemaName	字符串	未设置	此参数指定用作所有数据库对象的前缀的数据库模式。如果此参数为空，那么缺省情况是，将用于连接至数据库的用户标识用作前缀。请在管理控制台的数据源定义中设置此用户标识。如果对此参数指定了值，那么数据源中指定的用户标识必须有足够的权限来访问此模式的数据库对象。此参数必须与事件收集器的值匹配。
ReportAtSnapshotRange	整数/天	60	构建快照报告时，将对所有比快照的日期和时间条件旧的事件进行评估。 ReportAtSnapshotRange 定义快照报告中可以包括的事件的时间段。快照报告仅对此时间段内发出的事件进行评估。 如果此值过大，那么可能必须要处理非常大量的事件，从而导致生成报告的速度相当慢。请尝试将此值设置为业务环境中流程实例的最大持续时间。
ObserverCreateTables	布尔值		此参数指示当 EJB 第一次连接到数据库时，是否应该创建 Observer 模式。有效值为“true”和“false”。

更改观察器的配置参数

要更改观察器参数，请执行下列操作：

1. 启动用于设置观察器的工具，如第 240 页的『setupObserver 工具』所述。您将看到以下菜单：
 - 1) Install the Observer application
 - 2) Remove the Observer application and related objects
 - 3) Change configuration settings of an installed Observer

0) Exit Menu
2. 选择选项 3 以显示一组可以更改的参数：
 - 1) ReportAtSnapshotRange
 - 2) ObserverCreateTables
 - 3) ObserverSchemaName

0) Exit Menu
3. 选择要更改的参数的编号。将显示该参数的名称、描述、类型、单位和当前值。

4. 要更改指定的值，请输入新值，然后按 Enter 键。如果在未输入新值的情况下按 Enter 键，那么将返回参数列表。
5. 如果要更改另一个参数的值，请从步骤 3 重复。
6. 输入 0 以退出该列表。系统将询问您是否要保存更改。
7. 要保存所有更改，请输入 y，否则输入 n 以废弃所有更改。
8. 要激活更改，请重新启动 BPCObserver 应用程序。

setupEventCollector 工具

使用 setupEventCollector 以交互方式配置或删除业务流程编排器事件收集器应用程序、设置数据库以及管理数据库的用户定义函数。此工具使用 wsadmin 脚本。

位置

此工具在业务流程编排器的配置脚本子目录中：

在 Linux、UNIX 和 i5/OS 平台上：*install_root/ProcessChoreographer/config*。

在 Windows 平台上：*install_root\ProcessChoreographer\config*。

限制

- 在 Network Deployment 环境中，必须在 Deployment Manager 节点上启动此工具并使用 `-profileName` 选项来指定 Deployment Manager 概要文件。
- 此工具只有英文版。
- 在 i5/OS 上，必须使用 qshell 来运行此工具。

参数

```
[-conntype SOAP | RMI | JMS | NONE]
[-user userID -password password]
[-profileName profileName]
( [-node nodeName] [-server serverName] ) | ( -cluster clusterName )
[-remove [-silent]]
```

其中：

-conntype SOAP | RMI | JMS | NONE

wsadmin 工具使用的连接方式。在独立服务器环境中，仅当应用程序服务器未处于运行状态时，才需要指定 `-conntype NONE` 选项。在 Network Deployment 环境中，仅当 Deployment Manager 未处于运行状态时，才需要指定 `-conntype NONE` 选项。

-user *userID* -password *password*

如果已启用全局安全性，那么还需要提供有效的用户标识和密码供此工具使用。

-profileName *profileName*

如果所配置的不是缺省概要文件，那么请提供要配置的概要文件的名称。

-node *nodeName*

节点的名称。此参数是可选的。缺省值是本地节点。

-server *serverName*

服务器的名称。此参数是可选的。

-cluster *clusterName*

clusterName 是集群名。此参数是可选的。

-remove

如果要除去事件收集器应用程序，那么指定此选项。如果未指定此选项，那么缺省情况下将配置该应用程序。

-silent

此选项只能与 remove 选项配合使用。此选项将使此工具不输出任何提示。此参数是可选的。

注：如果未指定 -node、-server 和 -cluster 参数，那么在配置期间将提示您指定部署目标。

示例：启动此工具

要启动此工具，以处理名为 server1 的服务器，请输入下列其中一个命令。

在 **Linux** 和 **UNIX** 平台上：

```
setupEventCollector.sh -server server1
```

在 **i5/OS** 平台上：

```
setupEventCollector -server server1
```

在 **Windows** 平台上：

```
setupEventCollector.bat -server server1
```

您将看到命令菜单：

- 1) Prepare a database for the Event Collector and Observer
- 2) Install the Event Collector application
- 3) Remove the Event Collector application and related objects
- 4) Change configuration settings of an installed Event Collector
- 5) Drop the database schema of the Event Collector and Observer
- 6) Administer Observer related user-defined functions

- 0) Exit Menu

使用此工具

下列主题描述如何使用此工具来完成特定任务。

相关概念

第 222 页的『业务流程编排器观察器的用户定义函数』

借助业务流程编排器观察器，可以运行基于生成 SQL 查询的时间片或时间间隔的报告。要执行这些报告，业务流程编排器观察器要求在数据库中安装一些特定的用户定义函数（UDF）。

相关任务

第 224 页的『配置业务流程编排器事件收集器应用程序』

使用交互式工具或管理控制台来安装和配置事件收集器应用程序。

第 194 页的『使用 setupEventCollector 工具为业务流程编排器观察器准备 DB2 通用数据库』

本节描述如何在 Linux、UNIX 和 Windows 平台上使用菜单驱动的交互式工具和 createTablespace_Observer.sql 脚本来准备 DB2 通用数据库。

第 198 页的『使用 setupEventCollector 工具为业务流程编排器观察器准备 DB2 iSeries 版数据库』

本节描述如何从 i5/OS Qshell 环境中使用菜单驱动的交互式工具来准备 DB2 iSeries 数据库。

第 203 页的『在 USS 中为业务流程编排器观察器创建 DB2 z/OS 版数据库』
本节描述如何在 z/OS 机器上的 UNIX 系统服务 (USS) 中使用菜单驱动的交互工具以及 createTablespace_Observer.sql 脚本来创建 DB2 z/OS 版数据库。

第 206 页的『从远程系统为业务流程编排器观察器创建 DB2 z/OS 版数据库』
本节描述如何在 Linux、UNIX 或 Windows 系统上使用菜单驱动的交互式工具以及 createTablespace_Observer.sql 脚本来创建 DB2 z/OS 版数据库。

第 211 页的『使用 setupEventCollector 工具为业务流程编排器观察器准备 Derby 数据库』

本节描述如何在任何受支持平台上使用菜单驱动的交互工具 setupEventcollector 来准备 Derby 数据库。

第 215 页的『使用 setupEventCollector 工具为业务流程编排器观察器准备 Oracle 数据库』

本节描述如何在 Linux、UNIX 和 Windows 平台上使用菜单驱动的交互式工具和 createTablespace_Observer.sql 脚本来准备 Oracle 数据库。

第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』
通过使用 setupEventCollector 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数 (UDF) 与基于 SQL 的 UDF 之间进行切换。

第 247 页的『使用脚本除去业务流程编排器配置』
通过本任务从服务器或集群中除去业务流程管理器、人员任务管理器、业务流程编排器资源管理器和业务流程编排器观察器配置以及相关资源。

相关参考

第 234 页的『更改业务流程编排器观察器的配置参数』
对于启用验证功能以及提高性能而言，调整业务流程编排器观察器和事件收集器应用程序的配置参数至关重要。

setupObserver 工具

使用 setupObserver 以交互方式配置或除去业务流程编排器观察器应用程序或者更改配置参数。此工具使用 wsadmin 脚本。

位置

此工具在业务流程编排器的配置脚本子目录中：

在 Linux、UNIX 和 i5/OS 平台上：*install_root/ProcessChoreographer/config*。

在 Windows 平台上：*install_root\ProcessChoreographer\config*。

限制

- 在 Network Deployment 环境中，必须在 Deployment Manager 节点上启动此工具并使用 `-profileName` 选项来指定 Deployment Manager 概要文件。
- 此工具只有英文版。
- 在 i5/OS 上，必须使用 qshell 来运行此工具。

参数

```
[-conntype SOAP | RMI | JMS | NONE]
[-user userID -password password]
[-profileName profileName]
( [-node nodeName] [-server serverName] ) | ( -cluster clusterName )
[-remove [-silent]]
```

其中:

-conntype SOAP | RMI | JMS | NONE

wsadmin 工具使用的连接方式。在独立服务器环境中, 仅当应用程序服务器未处于运行状态时, 才需要指定 **-conntype NONE** 选项。在 Network Deployment 环境中, 仅当 Deployment Manager 未处于运行状态时, 才需要指定 **-conntype NONE** 选项。

-user *userID* -password *password*

如果已启用全局安全性, 那么还需要提供有效的用户标识和密码供此工具使用。

-profileName *profileName*

如果所配置的不是缺省概要文件, 那么请提供要配置的概要文件的名称。

-node *nodeName*

节点的名称。此参数是可选的。缺省值是本地节点。

-server *serverName*

服务器的名称。此参数是可选的。

-cluster *clusterName*

clusterName 是集群名。此参数是可选的。

-remove

如果要除去事件收集器应用程序, 那么指定此选项。如果未指定此选项, 那么缺省情况下将配置该应用程序。

-silent

此选项只能与 **remove** 选项配合使用。此选项将使此工具不输出任何提示。此参数是可选的。

注: 如果未指定 **-node**、**-server** 和 **-cluster** 参数, 那么在配置期间将提示您指定部署目标。

示例: 启动此工具

要启动此工具, 以处理名为 **server1** 的服务器, 请输入下列其中一个命令。

在 **Linux** 和 **UNIX** 平台上:

```
setupObserver.sh -server server1
```

在 **i5/OS** 平台上:

```
setupObserver -server server1
```

在 **Windows** 平台上:

```
setupObserver.bat -server server1
```

您将看到命令菜单:

- 1) Install the Observer application
- 2) Remove the Observer application and related objects
- 3) Change configuration settings of an installed Observer

- 0) Exit Menu

使用此工具

下列主题描述如何使用此工具来完成特定任务。

相关任务

第 230 页的『使用 setupObserver 工具配置业务流程编排器观察器』

本节描述如何使用菜单驱动的交互式工具来安装业务流程编排器观察器应用程序的实例，并描述如何对其进行配置以连接至特定事件收集器的数据源。

第 247 页的『使用脚本除去业务流程编排器配置』

通过本任务从服务器或集群中除去业务流程管理器、人员任务管理器、业务流程编排器资源管理器和业务流程编排器观察器配置以及相关资源。

相关参考

第 234 页的『更改业务流程编排器观察器的配置参数』

对于启用验证功能以及提高性能而言，调整业务流程编排器观察器和事件收集器应用程序的配置参数至关重要。

验证业务流程编排器观察器

在安装和配置业务流程编排器观察器之后，请验证观察器的工作是否正常。

开始之前

最初，业务流程编排器观察器数据库未包含任何内容。

过程

1. 执行将生成业务事件的操作，例如，使用业务流程编排器资源管理器来启动流程实例。
2. 在浏览器中，通过打开以下 URL 启动业务流程编排器观察器：`http://host:port/context_root/`。其中 *host* 是正在运行应用程序服务器的主机的名称，*port* 是应用程序服务器的端口号（缺省端口为 9080），*context_root* 通常是 `bpcobserver`。
3. 验证是否显示了您所预期的事件。如果看不到任何事件，那么请等待几分钟，重新启动事件收集器应用程序，然后刷新浏览器视图。

注： 如果使用缺省的 `BPCEventTransformerMaxWaitTime` 值和 `BPCEventTransformerToleranceTime` 值，那么在变换器被触发并且事件收集器表中的事件足够旧以便可供处理并可用之前，可能要经过长达 20 分钟的时间。有关这些参数的信息，包括更改方式以及用于进行测试目的的建议值，请参阅第 234 页的『更改业务流程编排器观察器的配置参数』。

4. 如果发生问题，请参阅第 585 页的『业务流程编排器观察器故障诊断』。

结果

业务流程编排器观察器运行正常。

激活业务流程编排器

配置业务流程编排器后，必须重新启动受影响的服务器或集群。

关于此任务

要激活业务流程编排器，请执行下列操作：

过程

1. 如果业务流程编排器被配置在服务器上，那么请重新启动该服务器。

2. 如果业务流程编排器被配置在集群中，那么请重新启动该集群。
3. 确保应用程序服务器的 `SystemOut.log` 文件未记录任何错误消息。在集群中，请检查该集群中所有应用程序服务器的日志。
4. 验证业务流程管理器和人员任务管理器应用程序是否已成功启动：在管理控制台中，选择 **应用程序** → **企业应用程序**，验证名称以 `BPEContainer_scope` 和 `TaskContainer_scope` 开头的应用程序是否已启动。

其中，`scope` 值为 `nodeName_serverName`（对于在应用程序服务器上配置的业务流程编排器）或者 `clusterName`（对于在集群中配置的业务流程编排器）。

结果

业务流程编排器处于运行状态。

下一步做什么？

您已准备好验证业务流程编排器是否工作正常。

验证业务流程编排器的运行是否正常

运行业务流程编排器安装验证应用程序。

过程

1. 通过使用管理控制台或 `wsadmin` 命令，安装 `install_root/installableApps/bpcivt.ear` 中的应用程序。安装企业应用程序后，该应用程序将处于“已停止”状态，它所包含的任何流程模板和任务模板都处于“已启动”状态。直到该应用程序启动后才能创建流程实例或任务实例。
2. 根据业务流程编排器的配置位置不同，请确保：
 - 应用程序服务器处于运行状态，或者
 - 至少其中一个集群成员处于运行状态
3. 确保数据库系统和消息传递服务处于运行状态。
4. 选择应用程序 `BPCIVTApp`，然后单击 **启动** 以启动该应用程序。
5. 验证应用程序的运行是否正常。使用 Web 浏览器，打开以下页面：
`http://app_server_host:port_no/bpcivt`

其中 `app_server_host` 是应用程序服务器所在主机的网络名，`port_no` 是安装 `bpcivt.ear` 文件时 `IVT Web` 模块所映射的虚拟主机使用的端口号。随系统配置的不同，端口号也会有所变化。您应该会看到指示操作成功的消息。

6. 可选：停止 `bpcivt` 应用程序，然后将其除去。
7. 如果发生错误，那么出错原因可能是：
 - 如果业务流程编排器无法访问数据库，那么请检查数据库系统是否处于运行状态、是否已正确配置所有数据库客户机以及是否已正确定义数据源。确保数据源的用户标识和密码有效。
 - 如果业务流程编排器无法读取输入队列，那么请检查消息传递服务是否处于运行状态，并确保已正确定义 `JMS` 提供程序和 `JMS` 资源。

结果

业务流程编排器配置的基本功能已生效。

下一步做什么？

如果已配置其他可选部件，例如业务流程编排器观察器、业务流程编排器资源管理器或人员目录提供程序，那么应该单独测试这些部件。

了解业务流程编排器的启动行为

本主题说明业务流程编排器直到所有企业应用程序启动后才可用的原因。

启动或重新启动业务流程编排器时，直到所有企业应用程序启动后才会开始处理内部队列中的消息。不可能对此行为进行更改。业务流程管理器和人员任务管理器在重新启动期间保持不可用的时间长度取决于启动所有企业应用程序所需的时间。这种行为旨在避免对相关企业应用程序未处于运行状态的任何流程进行导航。

如果在所有应用程序启动完成前开始处理内部队列中的消息，将导致 `ClassNotFoundException` 异常。

联合配置了业务流程编排器的独立节点

如果服务器未以开发方式运行，那么可以将独立概要文件中的服务器联合到新的 `Deployment Manager` 单元。

开始之前

`Deployment Manager` 处于运行状态，并且您知道它的主机名和端口号。在独立概要文件中的服务器上配置了业务流程编排器。必须能够从 `Deployment Manager` 单元以远程方式访问独立概要文件中的业务流程编排器数据库。因此，服务器不能基于使用嵌入式 `Derby` 数据库的业务流程编排器样本配置。并且，消息传递引擎数据库必须可供远程访问，即，它不能是嵌入式 `Derby` 数据库或 `FILESTORE`。

关于此任务

在独立服务器上运行了一个或多个包含业务流程或人员任务的应用程序，并且您要将此服务器联合到 `Network Deployment` 环境。

过程

1. 如果该节点包含大量应用程序，那么增大管理连接器的超时值。
2. 从命令行运行 `addNode` 命令并指定 `-includeapps` 和 `-includebuses` 选项。有关此命令以及可能发生的错误的详细信息，请参阅 `WebSphere Application Server Network Deployment` 信息中心中的 `addNode` 命令。例如，如果 `Deployment Manager` 的主机名为 `dmgr_host` 并使用端口 `dmgr_port`，那么输入以下命令：

```
addNode dmgr_host dmgr_port -includeapps -includebuses
```

例如，如果 `Deployment Manager` 的主机名为 `any.hostname.com` 并使用端口 `9043`，概要文件名为 `ProcSvr07`，您的用户标识为 `admin`，密码为 `secret`，那么输入以下命令：

```
addNode any.hostname.com 9043 -profileName ProcSvr07 -username admin -password secret -includeapps -includebuses
```

如果不符合任何先决条件，那么将显示错误消息。否则，该服务器将停止并联合到新的 Deployment Manager 单元中。

3. 启动服务器，以激活更改。
4. 如果无法访问该服务器上运行的业务应用程序，那么请使用 Deployment Manager 上的管理控制台，以确保该应用程序服务器的虚拟主机和别名定义与新单元匹配。

结果

现在，应用程序仍在原先的服务器上运行，但该服务器已包含在可以使用 Deployment Manager 进行管理的单元中。

下一步做什么？

根据需要，可以将该服务器提升为集群。

第 5 章 除去业务流程编排器配置

通过本任务除去业务流程容器、人员任务容器、业务流程编排器资源管理器、业务流程编排器观察器以及相关资源。

过程

1. 确保所有独立服务器、数据库和应用程序服务器（或者每个集群中的至少一个应用程序服务器）都处于运行状态。
2. 对于每个包含人员任务或业务流程的企业应用程序，请停止所有人员任务模板和所有业务流程模板。
3. 卸载所有包含人员任务或业务流程的企业应用程序。
4. 请执行下列其中一项操作：
 - 要除去业务流程编排器配置、业务流程编排器资源管理器、业务流程编排器观察器、事件收集器以及相关资源，请执行『使用脚本除去业务流程编排器配置』。
 - 如果要复用现有配置的某些部分，请执行第 250 页的『使用管理控制台除去业务流程编排器配置』。

结果

已除去业务流程编排器配置。

使用脚本除去业务流程编排器配置

通过本任务从服务器或集群中除去业务流程管理器、人员任务管理器、业务流程编排器资源管理器和业务流程编排器观察器配置以及相关资源。

开始之前

在可以除去业务流程编排器配置之前，必须停止所有流程模板和任务模板并删除所有流程实例和任务实例，然后停止并除去所有包含业务流程或人员任务的企业应用程序。

过程

1. 切换到业务流程编排器的 `config` 目录：

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\config
```

在 Linux、UNIX 和 i5/OS 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/config
```

2. 运行脚本 `bpeunconfig.jacl`。在下列情况下，还需要指定适当的选项：
 - 对于独立服务器，请停止应用程序服务器，然后使用 `-conntype NONE` 选项。此步骤确保未锁定任何 Derby 数据库，从而可以自动除去这些数据库。
 - 在 Network Deployment 环境中，请按如下方式运行此脚本：

- 如果 Deployment Manager 未处于运行状态，那么在 Deployment Manager 上运行此脚本并指定 `-conntype NONE` 选项。
- 如果 Deployment Manager 处于运行状态，那么停止要除去配置的应用程序服务器，然后运行此脚本，并且不要指定 `-conntype NONE` 选项。

在要除去业务流程编排器配置的应用程序服务器节点上运行此脚本时，此脚本可以自动删除所有本地 Derby 数据库。

- 如果已启用 WebSphere 管理安全性，那么还需指定用户标识和密码：
`-user userID -password password`
- 如果不是从缺省概要文件中除去配置，那么还需指定概要文件名：
`-profileName profileName`

选项	描述
对于运行 Linux 或 UNIX 的单服务器	输入命令： <code>install_root/bin/wsadmin.sh -f bpeunconfig.jacl -server Server -node Node [-deleteDB deleteDatabase]</code>
对于运行 Windows 的单服务器	输入命令： <code>install_root\bin\wsadmin.bat -f bpeunconfig.jacl -server Server -node Node [-deleteDB deleteDatabase]</code>
对于运行 i5/OS 的单服务器	输入命令： <code>install_root/bin/wsadmin -f bpeunconfig.jacl -server Server -node Node [-deleteDB deleteDatabase]</code>
对于运行 Linux 或 UNIX 的集群	输入命令： <code>install_root/bin/wsadmin.sh -f bpeunconfig.jacl -cluster Cluster</code>
对于运行 Windows 的集群	输入命令： <code>install_root\bin\wsadmin.bat -f bpeunconfig.jacl -cluster Cluster</code>
对于运行 i5/OS 的集群	输入命令： <code>install_root/bin/wsadmin -f bpeunconfig.jacl -cluster Cluster</code>

其中：

`userID` 用户标识。

`password`

该用户标识的密码。

`profileName`

所配置的概要文件的名称。如果正在配置缺省概要文件，那么此选项是可选的。

`Server` 应用程序服务器的名称。如果只有一个服务器，那么此参数是可选的。

`Node` 节点的名称。此参数是可选的。如果省略 `Node`，那么将使用本地节点。

Cluster 集群的名称。

deleteDatabase

此布尔值指定是否删除任何嵌入式 Derby 数据库和 FILESTORE 目录：

yes

no

要使用此选项，服务器不能处于运行状态。如果没有任何非嵌入式 Derby 数据库，并且使用了此选项，那么在运行此脚本后，可以跳转到 4 步骤。

如果省略了必需参数，那么将提示您进行输入。

3. 可选：删除业务流程编排器使用的数据库。

对于业务流程编排器数据库和消息传递数据库，下列各项适用：

- `bpeunconfig.jacl` 脚本将列示已除去的配置所使用的数据库。还会将数据库列表写入 `install_root/profiles/profileName/logs/bpeunconfig.log` 日志文件。可以使用此列表来标识要手动删除的数据库。
 - 将 Derby 数据库用作业务流程编排器数据库时，除非它被运行中的应用程序服务器锁定，否则 `bpeunconfig.jacl` 脚本可以选择除去该数据库。如果该数据库被锁定，那么请停止该服务器，然后使用 `-conntype NONE` 选项。
 - 当使用 Derby 数据库作为消息传递数据库时，除非它被运行中的应用程序服务器锁定，否则 `bpeunconfig.jacl` 脚本可以选择除去该数据库。如果该数据库被锁定，那么请停止该服务器，然后使用 `-conntype NONE` 选项。
 - 当使用 FILESTORE 作为业务流程编排器消息传递引擎的消息存储器时，如果指定了 `bpeunconfig.jacl` 脚本的 `-deleteDB yes` 选项，那么还将删除相关联的目录。
 - 要除去业务流程编排器观察器数据库，请启动用于设置事件收集器的工具（如第 238 页的『`setupEventCollector` 工具』所述）并选择 **Drop the database schema of the Event Collector and Observer** 选项。
4. 可选：检查 `bpeunconfig.log` 日志文件。此文件在 `profile_root` 目录的 `logs` 子目录中。
 5. 可选：如果曾经使用 WebSphere MQ，那么请删除业务流程编排器使用的队列管理器。
 6. 可选：手动撤销 `bpeunconfig.jacl` 未撤销的其余设置。由于 `bpeunconfig.jacl` 脚本无法确定其他组件是否仍然需要下列设置，所以不会将这些设置撤销：
 - 启用 `WorkAreaService`
 - 启用 `ApplicationProfileService`
 - 启用 `ObjectPoolService`
 - 启用 `StartupBeansService`
 - 启用 `CompensationService`
 - 启用 `WorkareaPartitionService`
 - 设置 WebSphere 变量

结果

已除去业务流程编排器应用程序以及相关资源（例如调度程序、数据源、侦听器端口、连接工厂、队列目标、激活规范、工作区分区、邮件会话和认证别名）。

使用工具除去业务流程编排器观察器和事件收集器

从服务器或集群中除去业务流程编排器观察器和事件收集器应用程序以及相关资源。

过程

1. 如果已配置业务流程编排器观察器，那么请运行第 240 页的『setupObserver 工具』并指定 `-remove` 选项。
2. 如果已配置业务流程编排器事件收集器应用程序，那么请运行第 238 页的『setupEventCollector 工具』并指定 `-remove` 选项。
3. 可选：如果已安装 Java 用户定义函数，那么请将其除去。
4. 可选：删除数据库。

结果

已除去业务流程编排器观察器和事件收集器应用程序。

使用管理控制台除去业务流程编排器配置

通过本任务除去部分或全部业务流程编排器配置，其中包括业务流程编排器资源管理器、业务流程编排器观察器以及相关资源。

开始之前

在可以除去业务流程编排器配置前，必须停止所有流程模板和任务模板并删除所有任务实例和流程实例，然后卸载所有包含业务流程或人员任务的企业应用程序。

过程

1. 卸载业务流程编排器企业应用程序。
 - a. 显示企业应用程序。

在管理控制台中，选择应用程序 → 企业应用程序。

- b. 标识业务流程编排器安装版本的作用域。

查找名称以下列内容开头的应用程序：

- `BPEContainer_scope` 是业务流程管理器应用程序。
- `TaskContainer_scope` 是人员任务管理器应用程序。
- `BPCExplorer_scope` 是业务流程编排器资源管理器应用程序。

其中，`scope` 值随配置的不同而有所变化：

- 如果业务流程编排器最初被配置在应用程序服务器上，那么 `scope` 值为 `nodeName_serverName` - 即使之后将该服务器提升为集群亦如此。
 - 如果业务流程编排器被配置在集群中，那么 `scope` 值为 `clusterName`。
- c. 可选：如果已配置业务流程编排器，那么请卸载业务流程管理器和人员任务管理器应用程序。选择 `BPEContainer_scope` 和 `TaskContainer_scope`，然后单击卸载 → 确定 → 保存。
 - d. 可选：如果已配置业务流程编排器资源管理器，那么请将其卸载。

- 如果已使用缺省上下文根 `/bpc`，那么请选择 `BPCExplorer_scope`，然后单击 **卸载** → **确定** → **保存**。
- 否则，选择 `BPCExplorer_scope_context_root`，然后单击 **卸载** → **确定** → **保存**。

注： 卸载业务流程编排器观察器和事件收集器应用程序，如第 254 页的 10 步骤所述。

2. 除去下列全部或任何您不想重复使用的资源：

- a. 可选： 查找缺省名称为 `BPEDataSourcedbType` 的业务流程编排器数据源，记录它的名称、可能有的相关认证数据别名以及 Java 命名和目录接口 (JNDI) 名称，然后将其除去 (缺省名称为 `jdbc/BPEDB`)。

要查找数据源，请执行下列操作：

- 1) 单击 **资源** → **JDBC** → **数据源**。
- 2) 对于 **作用域**，请选择配置了业务流程编排器的服务器或集群。

- b. 可选： 对于除 Derby 数据库以外的数据库，请除去 2 步骤中标识的数据源的 JDBC 提供程序，除非该提供程序包含您仍需要的其他数据源。 单击 **资源** → **JDBC** → **JDBC 提供程序**，选择数据库的 JDBC 提供程序，然后单击 **删除**。

注： 如果业务流程编排器配置使用嵌入式 Derby 数据库的内置缺省 JDBC 提供程序，那么无法删除此 JDBC 提供程序。

- c. 可选： 除去相应的连接工厂和队列。
- 对于缺省消息传递，在除去连接工厂之前，请记录它们的相关认证数据别名。然后，除去 JMS 连接工厂和 JMS 队列。
 - 1) 单击 **资源** → **JMS** → **连接工厂**。对于 **作用域**，请选择配置了业务流程编排器的服务器或集群。然后，选择连接工厂并单击 **删除**。
 - 2) 单击 **资源** → **JMS** → **队列**。对于 **作用域**，请选择配置了业务流程编排器的服务器或集群。然后，选择队列并单击 **删除**。
 - 对于 WebSphere MQ，请除去 JMS 队列连接工厂和 JMS 队列。
 - 1) 单击 **资源** → **JMS** → **队列连接工厂**。对于 **作用域**，请选择配置了业务流程编排器的服务器或集群。然后，选择连接工厂并单击 **删除**。
 - 2) 单击 **资源** → **JMS** → **队列**。对于 **作用域**，请选择配置了业务流程编排器的服务器或集群。然后，选择队列并单击 **删除**。

对于业务流程容器而言，JNDI 名称通常如下所示：

```
jms/BPECF
jms/BPEFCF
jms/BFMJMSReplyCF
jms/BPEIntQueue
jms/BPERetQueue
jms/BPEHldQueue
jms/BFMJMSAPIQueue
jms/BFMJMScallbackQueue
jms/BFMJMSReplyQueue
```

对于人员任务容器而言，JNDI 名称通常如下所示：

```
jms/HTMCF
```

jms/HTMIntQueue
jms/HTMHldQueue

d. 可选: 如果正在使用 WebSphere 缺省消息传递作为 JMS 提供程序, 那么请除去激活规范。

1) 单击**资源** → **JMS** → **激活规范**。对于**作用域**, 请选择配置了业务流程编排器的服务器或集群。

2) 除去下列激活规范:

BPEInternalActivationSpec
BFMJMSAS
HTMInternalActivationSpec

e. 可选: 如果正在使用 WebSphere MQ 作为 JMS 提供程序, 那么请除去服务器的侦听器端口。

1) 单击**服务器** → **应用程序服务器** → **serverName**。

2) 在“通信”中, 单击**消息传递** → **消息侦听器服务** → **侦听器端口**。

3) 在应用程序服务器窗格中, 除去下列侦听器端口:

BPEInternalListenerPort
BPEHoldListenerPort
HTMInternalListenerPort

如果业务流程编排器被配置在集群中, 那么请对该集群中的每个成员重复此步骤。

f. 可选: 删除认证数据别名。

1) 单击**安全性** → **安全管理、应用程序和基础结构**, 然后在**认证**部分中展开 **Java 认证和授权服务**, 接着单击 **J2C 认证数据**。

2) 如果第 251 页的 2 步骤中标识的数据源具有认证数据别名, 那么除去该别名。如果业务流程编排器配置不是从 V6.0.x 迁移而来, 那么名称取决于部署目标, 如下所示:

- 如果业务流程编排器被配置在节点 *nodeName* 上名为 *serverName* 的服务器上, 那么名称通常是 `BPCDB_nodeName.serverName_Auth_Alias`。
- 如果业务流程编排器被配置在名为 *clusterName* 的集群中, 那么名称通常是 `BPCDB_clusterName_Auth_Alias`。

3) 如果第 251 页的 2c 步骤中标识的任何连接工厂具有认证数据别名, 那么请小心地除去该别名:

- 如果业务流程编排器配置不是从 V6.0.x 迁移而来, 那么名称是 `BPC_Auth_Alias` 并且由 Network Deployment 环境中的所有业务流程编排器配置共享。

警告: 仅当正在除去最后一个业务流程编排器配置时才应该除去此认证别名, 否则, 其余业务流程编排器配置将停止运行。

- 如果业务流程编排器配置是从 V6.0.x 迁移而来, 那么名称通常是 `cellName/BPEAuthDataAliasJMS_scope`, 其中 *cellName* 是单元名, *scope* 指示部署目标。您可以除去此认证别名, 而不会影响其他业务流程编排器配置。

g. 可选: 除去数据源 JNDI 名称的调度程序配置。

1) 单击**资源** → **调度程序**。

2) 对于**作用域**, 请选择配置了业务流程编排器的服务器或集群。

- 3) 在调度程序窗格中，记录工作管理器的 JNDI 名称，然后选择并删除调度程序 BPEScheduler。
- h. 可选： 除去工作管理器。
 - 1) 单击资源 → 异步 Bean → 工作管理器。
 - 2) 对于作用域，请选择配置了业务流程编排器的服务器或集群。
 - 3) 在工作管理器窗格中，选择并删除步骤 第 252 页的 2g 中记录的 JNDI 名称的相应工作管理器。
 - 4) 并且，请删除 JNDI 名称为 `wm/BPENavigationWorkManager` 的工作管理器。
- i. 可选： 除去工作区分区。
 - 1) 单击服务器 → 应用程序服务器 → **serverName**。
 - 2) 在容器设置部分中展开业务流程服务，然后单击工作区分区服务。
 - 3) 在应用程序服务器窗格中，选择并删除工作区分区 BPECompensation。

如果业务流程编排器被配置在集群中，那么请对该集群中的每个成员重复此步骤。
- j. 可选： 除去邮件会话。
 - 1) 单击资源 → 邮件 → 邮件提供程序。
 - 2) 对于作用域，选择单元=**cellName**，其中 *cellName* 是单元名。
 - 3) 单击内置邮件提供程序。
 - 4) 在其他属性部分中，选择邮件会话。
 - 5) 选择并删除 `HTMailSession_scope`，其中 *scope* 是第 250 页的 1b 步骤中标识的作用域。
3. 可选： 如果将 WebSphere 缺省消息传递用于业务流程编排器，那么可以删除总线成员、总线和数据源：
 - a. 单击服务集成 → 总线 → **BPC.cellName.Bus**，然后在拓扑部分中单击消息传递引擎。
 - b. 选择消息传递引擎：
 - 如果业务流程编排器被配置在服务器上，那么请选择 **nodeName.serverName-BPC.cellName.Bus**。
 - 如果业务流程编排器被配置在集群中，那么请选择 **clusterName-BPC.cellName.Bus**。

注： 如果已将业务流程编排器配置为使用远程消息传递引擎，那么 *nodeName.serverName* 或 *clusterName* 可能与配置了该业务流程编排器的部署目标的名称不匹配。
 - c. 在其他属性中，选择消息存储器。
 - 如果消息存储器类型为 DATASTORE，那么请记录数据源的 JNDI 名称。在服务器上，数据源的 JNDI 名称通常是 `jdbc/com.ibm.ws.sib/nodeName.serverName-BPC.cellName.Bus`。在集群中，数据源的 JNDI 名称通常是 `jdbc/com.ibm.ws.sib/clusterName-BPC.cellName.Bus`。
 - 如果消息存储器类型为 FILESTORE，那么请记录日志、永久存储器和临时存储器的路径。
 - d. 转到服务集成 → 总线 → **BPC.cellName.Bus**，在拓扑部分中单击总线成员，然后除去由下列其中一个名称标识的总线成员：

- *nodeName:serverName* (如果业务流程编排器被配置在服务器上)
 - *clusterName* (如果业务流程编排器被配置在集群中)
- e. 可选: 如果已除去总线 *BPC.cellName.Bus* 的最后一个成员, 那么还可以除去该总线。
 - f. 如果步骤 第 253 页的 3c 中记录的消息存储器类型为 *DATASTORE*, 那么请单击**资源** → **JDBC** → **数据源**。消息传递引擎的作用域可能与配置了业务流程编排器的部署目标不同。必要时, 请尝试使用其他作用域并查找步骤 第 253 页的 3c 中记录的 *JNDI* 名称。如果该数据源用于 *Derby* 数据库, 那么请记录该数据库的文件系统路径。如果业务流程编排器被配置在集群中, 那么请对该集群中的每个成员重复此步骤。
4. 删除 *BPC_REMOTE_DESTINATION_LOCATION* 变量。单击**环境** → **WebSphere 变量**。对于**作用域**, 请选择业务流程编排器配置所在的部署目标, 然后选择并删除 *BPC_REMOTE_DESTINATION_LOCATION* 变量。
 5. 重新启动该应用程序服务器或集群。
 6. 单击**保存**以将所有更改保存到主配置。
 7. 可选: 删除业务流程编排器数据库。
 8. 可选: 如果正在使用 *WebSphere MQ*, 那么请删除业务流程编排器所使用的队列管理器。
 9. 如果将 *WebSphere* 缺省消息传递用于业务流程编排器, 那么请删除消息传递引擎的消息存储器; 这是因为, 无法重复使用此消息存储器。
 - a. 如果第 253 页的 3c 步骤中记录的消息存储器类型为 *FILESTORE*, 那么请除去所记录的日志、永久存储器和临时存储器目录。
 - b. 如果第 253 页的 3c 步骤中记录的消息存储器类型为 *DATASTORE*, 那么请除去该数据源所指向的数据库。如果这是 *Derby* 数据源, 那么请删除 3f 步骤中记录的文件系统路径。通常, *Derby* 数据库位置如下所示:
 - 在 *Linux*、*UNIX* 和 *i5/OS* 平台上:


```
profile_root/databases/com.ibm.ws.sib/  
nodeName.serverName-BPC.cellName.Bus
```
 - 在 *Windows* 平台上:


```
profile_root\databases\com.ibm.ws.sib\  
nodeName.serverName-BPC.cellName.Bus
```
 10. 可选: 如果已配置业务流程编排器观察器, 那么请执行下列操作:
 - a. 对于每个观察器应用程序实例, 执行第 256 页的『使用管理控制台除去业务流程编排器观察器』。
 - b. 对于每个事件收集器应用程序实例, 执行第 255 页的『使用管理控制台除去业务流程编排器事件收集器』。

结果

已除去业务流程编排器配置。

使用管理控制台除去业务流程编排器事件收集器

通过本任务除去业务流程编排器观察器所需的业务流程编排器事件收集器配置以及相关资源。

过程

1. 显示企业应用程序。

在管理控制台中，选择应用程序 → 企业应用程序。

2. 卸载业务流程编排器事件收集器应用程序。选中 `BPCCollector_scope` 的复选框，然后单击卸载 → 确定。其中，`scope` 标识配置了事件收集器的服务器或集群。

3. 删除目标队列：

- a. 单击服务集成 → 总线 → **CommonEventInfrastructure_Bus** 。

- b. 在目标资源中，单击目标。

- c. 选择下列目标队列：

- `BPCCEIConsumerQueueDestination_scope`
- `BPCTransformerQueueDestination_scope`

其中，`scope` 标识配置了事件收集器的服务器或集群。

- d. 单击删除。

4. 删除 JMS 队列连接工厂：

- a. 单击资源 → **JMS** → 队列连接工厂。

- b. 对于作用域，选择配置了事件收集器的服务器或集群。

- c. 选中 `BPCCEIConsumerQueueConnectionFactory` 的复选框。

- d. 单击删除。

5. 删除 JMS 队列：

- a. 单击资源 → **JMS** → 队列。

- b. 选中下列队列的复选框：

- `BPCCEIConsumerQueue_scope`
- `BPCTransformerQueue_scope`

- c. 单击删除。

6. 删除 JMS 激活规范：

- a. 单击资源 → **JMS** → 激活规范。

- b. 选中下列激活规范的复选框：

- `BPCCEIConsumerActivationSpec`
- `BPCTransformerActivationSpec`

- c. 单击删除。

7. 删除 `BFMEvents` 的具有服务器作用域的事件概要文件组：

- a. 单击服务集成 → 公共事件基础结构 → 事件服务。

- b. 在其他属性中，单击事件服务。

- c. 单击缺省公共事件基础结构事件服务器。

- d. 在其他属性中，单击事件组。

- e. 选中 `BFMEvents` 的复选框。

- f. 单击删除。
8. 如果配置是从 V6.0.2 迁移而来，那么请删除认证数据别名：
 - a. 单击安全性 → 安全管理、应用程序和基础结构 → 认证 → Java 认证和授权服务 → J2C 认证数据。
 - b. 选择 BPCEventCollectorJMSAuthenticationAlias_scope。
 - c. 单击删除。
9. 单击保存以将更改保存至主配置。
10. 通过运行下列脚本，删除业务流程编排器观察器所使用的数据库、模式和表空间。在 Windows 平台上，这些脚本在 `install_root\dbscripts\ProcessChoreographer\database_type` 目录中；在 Linux、UNIX 和 i5/OS 平台上，这些脚本在 `install_root/dbscripts/ProcessChoreographer/database_type` 目录中：
 - `dropSchema_Observer.sql`
 - `dropTablespace_Observer.sql`

结果

已除去业务流程编排器事件收集器配置。

使用管理控制台除去业务流程编排器观察器

通过本任务除去业务流程编排器观察器配置以及相关资源。

过程

1. 显示企业应用程序。

在管理控制台中，选择应用程序 → 企业应用程序。

2. 找到业务流程编排器观察器实例。

查找名称以 `BPCObserver_scope` 开头的应用程序。

- 如果业务流程编排器观察器安装在应用程序服务器上，那么 `scope` 值为 `nodeName_serverName`。
- 如果业务流程编排器观察器安装在集群中，那么 `scope` 值为 `clusterName`。

注：如果上下文根不是缺省值 `/bpcobserver`，那么应用程序名后面还将追加上下文根 `_contextRoot`。

3. 卸载业务流程编排器观察器应用程序。选择要删除的应用程序实例，然后单击卸载 → 确定 → 保存。

结果

已除去业务流程编排器观察器配置。

第 3 部分 管理

第 6 章 管理业务流程编排器

可以使用管理控制台或脚本来管理业务流程编排器。

使用管理控制台管理业务流程编排器

本节描述可以使用管理控制台执行的管理操作。

管理服务器的补偿服务

使用管理控制台将补偿服务设置成随应用程序服务器启动而自动启动，并指定恢复日志的位置和最大大小。

关于此任务

在应用程序服务器上运行业务流程时，必须在该服务器上启动补偿服务。补偿服务用于管理流程完成前在大量事务中进行的更新。设置新的应用程序服务器时，缺省情况下将启用补偿服务。

您可以使用管理控制台来查看和更改应用程序服务器的补偿服务属性。

过程

1. 显示管理控制台。
2. 在导航窗格中，单击**服务器** → **应用程序服务器** → **server_name**。
3. 在“配置”选项卡的“容器设置”中，单击**容器服务** → **补偿服务**。此操作将显示包含补偿服务属性的面板。确保选中**在服务器启动时启用服务**复选框。如果在集群中运行业务流程，那么请对该集群中的每个服务器启用补偿服务。
4. 可选：根据需要，更改补偿服务的属性。
5. 单击**确定**。
6. 要保存配置，请在管理控制台窗口的消息框中单击**保存**。

使用管理控制台查询和重放已失败的消息

本节描述如何检查和重放未能处理的业务流程或人员任务的消息。

关于此任务

如果处理消息时发生了问题，那么该消息将移至保留队列或挂起队列。本任务描述如何确定是否存在已失败的消息以及如何将那些消息再次发送至内部队列。

过程

1. 要检查挂起队列和保留队列中的消息数：
 - a. 单击**服务器** → **应用程序服务器** → **server_name**，或者，如果业务流程编排器被配置在集群中，那么请单击**服务器** → **集群** → **cluster_name**。
 - b. 在**配置**选项卡上的“业务集成”中，单击**业务流程编排器**。选择下列其中一个选项：

- 对于业务流程，单击**业务流程管理器**。
- 对于人员任务，单击**人员任务管理器**。

挂起队列和保留队列中的消息数将显示在**运行时**选项卡的**常规属性**中。

2. 如果挂起队列和保留队列包含消息，那么可以将那些消息移至内部工作队列。

单击下列其中一个选项：

- 对于业务流程：**重放挂起队列**或**重放保留队列**
- 对于人员任务：**重放挂起队列**

注：如果已启用 WebSphere 管理安全性，那么重放按钮仅对具有操作员权限的用户可视。

结果

业务流程编排器将尝试再次处理所重放的所有消息。

相关概念

第 26 页的『从基础结构故障恢复』

业务流程管理器提供了用于处理基础结构临时故障的功能。

刷新失败消息数

使用管理控制台刷新业务流程或人员任务的失败消息数。

关于此任务

执行刷新前，显示的挂起队列消息数和保留队列消息数以及消息异常数将保持不变。本任务描述如何更新和显示那些队列中的消息数以及消息异常数。

过程

1. 选择适当的应用程序服务器。

单击**服务器** → **应用程序服务器** → **server_name**，或者，如果业务流程编排器被配置在集群中，那么请单击**服务器** → **集群** → **cluster_name**。

2. 刷新消息数。

- a. 在**配置**选项卡上的“业务集成”中，单击**业务流程编排器**。选择下列其中一个选项：
 - 对于业务流程，单击**业务流程管理器**。
 - 对于人员任务，单击**人员任务管理器**。
- b. 在**运行时**选项卡上，单击**刷新消息计数**。

结果

常规属性将显示下列更新后的值：

- 对于业务流程：挂起队列中的消息数和保留队列中的消息数
- 对于人员任务：挂起队列中的消息数
- 如果访问这些队列时发生了任何异常，那么“消息异常”字段将显示消息文本。

下一步做什么？

在此页面上，还可以重放这些队列中的消息。

相关概念

第 26 页的『从基础结构故障恢复』

业务流程管理器提供了用于处理基础结构临时故障的功能。

使用管理控制台刷新人员查询结果

人员查询的结果是静态的。您可以使用管理控制台来刷新人员查询。

关于此任务

业务流程编排器将已根据人员目录（例如轻量级目录访问协议（LDAP）服务器）进行求值的人员查询的结果高速缓存在运行时数据库中。如果人员目录发生更改，那么可以强制再次对人员分配进行评估。

过程

要刷新人员查询，请执行下列操作：

1. 单击**服务器** → **应用程序服务器** → **server_name**，或者，如果业务流程编排器被配置在集群中，那么请单击**服务器** → **集群** → **cluster_name**。
2. 在**配置**选项卡上的“业务集成”中，单击**业务流程编排器** → **人员任务管理器**。
3. 在**运行时**选项卡上，单击**刷新人员查询**。这将刷新所有人员查询。

注：如果已启用 WebSphere 管理安全性，那么“刷新”按钮仅对具有操作员权限的用户可视。

以此方式刷新人员查询结果会导致应用程序和数据库的负载较高。请考虑使用下列替代方法。

结果

相关概念

第 76 页的『管理人员分配条件和人员解析结果』

与任务授权角色相关联的人员分配条件在所部署任务模板或任务实例的整个生存期内有效。

相关任务

第 279 页的『使用管理命令刷新人员查询结果』

人员查询的结果是静态的。您可以使用管理命令来刷新人员查询。

第 263 页的『使用刷新守护程序刷新人员查询结果』

如果要对所有已到期的人员查询结果自动执行定期刷新，请使用本方法。

使用管理控制台启用公共基本事件和审计跟踪

通过本任务启用要作为公共基本事件发送到公共事件基础结构并且/或者存储在审计跟踪中的业务流程编排器事件。

关于此任务

可以在“配置”选项卡上永久地更改业务流程管理器或人员任务管理器的状态观察器设置，也可以在“运行时”选项卡上临时地进行更改。您在“配置”或“运行时”选项卡上进行

的选择将影响相应容器中执行的所有应用程序。对于同时影响业务流程管理器 and 人员任务管理器的更改，必须分别为它们进行设置更改。

使用管理控制台更改已配置的日志记录基础结构

通过本任务更改审计日志的状态观察器日志记录或配置的公共事件基础结构日志记录。

关于此任务

在“配置”选项卡上进行的更改将在服务器下次启动时生效。每当服务器启动时，选择的设置都将生效。

请按如下方式对配置进行更改：

过程

1. 显示业务流程管理器或人员任务管理器窗格。
 - a. 单击**服务器** → **应用程序服务器** → **server_name**，或者，如果业务流程编排器被配置在集群中，那么请单击**服务器** → **集群** → **cluster_name**。
 - b. 在**配置**选项卡上的“业务集成”中，单击**业务流程编排器**。选择下列其中一个选项：
 - 对于业务流程，单击**业务流程管理器**。
 - 对于人员任务，单击**人员任务管理器**。
2. 在**配置**选项卡上的“常规属性”部分中，选择要启用的日志记录功能。两个状态观察器相互独立：可以启用或禁用其中一个，也可以将它们全都启用或禁用。
 - 启用公共事件基础结构日志记录**
要根据公共事件基础结构来发出事件，请选中此复选框。
 - 启用审计日志记录**
要将审计日志事件存储在业务流程编排器数据库的审计跟踪表中，请选中此复选框。
3. 接受更改。
 - a. 单击**确定**。
 - b. 在“消息”框中，单击**保存**。

结果

已按要求对状态观察器进行设置。所作的更改将在服务器重新启动后生效。

下一步做什么？

重新启动服务器，以使更改生效。如果业务流程编排器被配置在集群中，那么请重新启动该集群。

使用管理控制台来配置会话的日志记录基础结构

通过本任务更改审计日志的状态观察器日志记录或会话的公共事件基础结构日志记录。

关于此任务

在“运行时”选项卡上进行的更改将立即生效。选择的设置将在服务器下次启动前保持有效。

请按如下方式对会话基础结构进行更改：

过程

1. 显示业务流程管理器或人员任务管理器窗格。
 - a. 单击**服务器** → **应用程序服务器** → **server_name**，或者，如果业务流程编排器被配置在集群中，那么请单击**服务器** → **集群** → **cluster_name**。
 - b. 在**配置**选项卡上的“业务集成”中，单击**业务流程编排器**。选择下列其中一个选项：
 - 对于业务流程，单击**业务流程管理器**。
 - 对于人员任务，单击**人员任务管理器**。
2. 在**运行时**选项卡上的“常规属性”部分中，选择要启用的日志记录功能。两个状态观察器相互独立：可以启用或禁用其中一个，也可以将它们全都启用或禁用。

启用公共事件基础结构日志记录

要根据公共事件基础结构来发出事件，请选中此复选框。

启用审计日志记录

要将审计日志事件存储在业务流程编排器数据库的审计跟踪表中，请选中此复选框。

3. 单击**确定**以接受更改。

结果

已按要求对状态观察器进行设置。

使用刷新守护程序刷新人员查询结果

如果要对所有已到期的人员查询结果自动执行定期刷新，请使用本方法。

关于此任务

人员查询由指定的人员目录提供程序解析。结果存储在业务流程编排器数据库中。为了提高授权性能，对检索到的查询结果进行了高速缓存。当调用人员查询刷新守护程序时，将检查高速缓存内容是否最新。

为了保持人员查询结果最新，产品提供了一个守护程序来定期刷新已到期的人员查询结果。此守护程序将刷新高速缓存中所有已到期的人员查询结果。

过程

1. 打开人员任务管理器的定制属性页面：
 - a. 单击**服务器** → **应用程序服务器** → **server_name**，或者，如果业务流程编排器被配置在集群中，那么请单击**服务器** → **集群** → **cluster_name**。
 - b. 在**配置**选项卡上的“业务集成”中，单击**业务流程编排器** → **人员任务管理器**。
 - c. 选择下列其中一个选项：
 - 要永久更改设置，请单击**配置**选项卡。所作的更改将在应用程序服务器重新启动后生效。

- 要临时更改设置，请单击**运行时**选项卡。所作的更改将立即生效，但将在应用程序服务器下次重新启动时被复位。
2. 在**人员查询刷新时间表**字段中，使用 WebSphere CRON 日历所支持的语法来输入时间表。此值确定守护程序刷新已到期人员查询结果的时间。缺省值是“0 0 1 * * ?”，即每天凌晨一点刷新一次。
 3. 在**人员查询结果超时**字段中，输入以秒计的新值。此值确定人员查询结果被视为有效的时间长度。这段时间过后，该人员查询结果将被视为不再有效，该守护程序下次运行时将刷新该人员查询。缺省值是 1 小时。
 4. 单击**确定**。
 5. 保存更改。要使您在“配置”选项卡上所作的更改生效，请重新启动应用程序服务器。

新的到期时间值将仅应用于新的人员查询，而不会应用于现有的人员查询。

相关概念

第 76 页的『[管理人员分配条件和人员解析结果](#)』

与任务授权角色相关联的人员分配条件在所部署任务模板或任务实例的整个生存期内有效。

使用脚本管理业务流程编排器

本节描述可以使用脚本执行的管理操作。

相关信息

第 233 页的『[使用脚本对业务流程编排器启用日志记录](#)』

本节描述如何使用 `setStateObserver.py` 脚本对业务流程编排器启用或禁用公共事件基础结构 (CEI) 或审计事件。

使用管理命令删除审计日志条目

使用管理命令来删除业务流程管理器的某些或全部审计日志条目。

开始之前

在开始执行本过程之前，必须符合下列条件：

- 删除审计日志条目时所借助的应用程序服务器必须处于运行状态。即，由于需要使用服务器连接，因此不能使用 `wsadmin` 的 `-conntype none` 选项。
- 如果业务流程编排器被配置在集群中，那么必须至少有一个集群成员处于运行状态。
- 如果已启用 WebSphere 管理安全性，那么您必须具有操作员权限。

关于此任务

可以使用 `deleteAuditLog.py` 脚本从数据库中删除业务流程管理器的审计日志条目。

过程

1. 切换到管理脚本所在的工作流程编排器子目录。

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 和 i5/OS 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/admin
```

2. 删除审计日志表中的条目。

在 Windows 平台上，输入下列一个或多个命令。这些命令之间的差别已突出显示：

```
install_root\bin\wsadmin -lang jython -f deleteAuditLog.py  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root\bin\wsadmin -lang jython -f deleteAuditLog.py  
-node nodeName  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root\bin\wsadmin -lang jython -f deleteAuditLog.py  
-cluster cluster_name  
[-profileName profileName]  
[options]
```

在 Linux 和 UNIX 平台上，输入下列其中一个或多个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin.sh -lang jython -f deleteAuditLog.py  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteAuditLog.py  
-node nodeName  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteAuditLog.py  
-cluster cluster_name  
[-profileName profileName]  
[options]
```

在 i5/OS 平台上，输入下列其中一个或多个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin -lang jython -f deleteAuditLog.py  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root/bin/wsadmin -lang jython -f deleteAuditLog.py  
-node nodeName  
-server server_name  
[-profileName profileName]  
[options]
```

```
install_root/bin/wsadmin -lang jython -f deleteAuditLog.py  
-cluster cluster_name  
[-profileName profileName]  
[options]
```

其中：

-cluster *cluster_name*

集群的名称。如果业务流程管理器被配置在 WebSphere 集群中，那么此选项是必需的。

-node *nodeName*

在指定服务器名时，此选项是可选的。此名称用于标识节点。缺省值是本地节点。

-server *server_name*

服务器的名称。如果未指定集群名，那么此选项是必需的。

-profileName *profileName*

用户定义的概要文件的名称。如果未使用缺省概要文件，那么应该指定此选项。

可用的 options 是:

-all

删除数据库中的所有审计日志条目。删除操作通过多个事务完成。每个事务都删除 *slice* 参数中指定的条目数或缺省条目数。

-time *timestamp*

删除所有早于 *timestamp* 所指定时间的审计日志条目。使用的时间是全球标准时间 (UTC)。其格式必须是: YYYY-MM-DD['T'HH:MM:SS]。如果仅指定了年月日，那么小时、分钟和秒设置为 00:00:00。

-time 与 *-processtime* 选项互斥。

-processtime *timestamp*

删除所有在 *timestamp* 所指定时间之前完成的进程的审计日志条目。请使用 *-time* 参数的时间格式。

-time 与 *-processtime* 选项互斥。

-slice *size*

size 与 *-all* 参数配合使用，此参数指定每个事务处理的条目数。最优值取决于数据库系统的可用日志大小。此值越大，需要的事务就越少，但可能会超出数据库日志空间。较小的值可能会导致此脚本完成删除操作的时间较长。*slice* 参数的缺省大小是 250。

注: 不推荐使用 JACL 版本的清理未用人员查询脚本 `deleteAuditLog.jacl`。此脚本在 `ProcessChoreographer` 目录的 `util` 子目录中，参数与上述参数相同，但不能指定 `-lang jython` 选项。

删除无效的流程模板

使用管理命令从业务流程编排器数据库中删除不再有效的流程模板。

开始之前

在开始执行本过程之前，要删除的模板所在的应用程序服务器必须处于运行状态。即，由于需要使用服务器连接，因此不能使用 `wsadmin` 的 `-conntype none` 选项。运行此命令并不需要任何特权，即使 WebSphere 管理安全性处于启用状态亦如此。如果业务流程编排器被配置在集群中，那么必须至少有一个集群成员处于运行状态。

关于此任务

对于数据库中那些未包含在 WebSphere 配置库中任何对应有效应用程序中的模板，可以使用 `deleteInvalidProcessTemplate.py` 脚本将它们以及所有属于它们的对象除去。如果用

户取消了应用程序的安装版本或者未将其存储在配置库中，那么会发生需要除去这些模板及其对象的情况。这些模板通常不起作用。它们不会显示在业务流程编排器资源管理器中。

在一些很罕见的情况下，无法对这些模板进行过滤。在这些情况下，必须使用下列脚本从数据库中除去这些模板。

不能使用这些脚本从数据库中除去有效应用程序的模板。将对此情况进行检查，如果相应的应用程序有效，那么将抛出 `ConfigurationError` 异常。

过程

1. 切换到管理脚本所在的业务流程编排器子目录。

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 和 iOS 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/admin
```

2. 从数据库中删除不再有效的业务流程模板。

在 Windows 系统上，要删除不再有效的业务流程模板，请输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root\bin\wsadmin -lang jython -f deleteInvalidProcessTemplate.py  
-server server_name  
-templateName templateName  
-validFrom validFromString  
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f deleteInvalidProcessTemplate.py  
-server server_name  
-node nodeName  
-templateName templateName  
-validFrom validFromString  
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f deleteInvalidProcessTemplate.py  
-cluster cluster_name  
-templateName templateName  
-validFrom validFromString  
[-profileName profileName]
```

在 Linux 和 UNIX 系统上，要删除不再有效的业务流程模板，请输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidProcessTemplate.py  
-server server_name  
-templateName templateName  
-validFrom validFromString  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidProcessTemplate.py  
-server server_name  
-node nodeName  
-templateName templateName  
-validFrom validFromString  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidProcessTemplate.py  
-cluster cluster_name
```



```
-templateName templateName  
-validFrom validFromString  
[-profileName profileName]
```

在 i5/OS 系统上，要删除不再有效的业务流程模板，请输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin -lang jython -f deleteInvalidProcessTemplate.py  
-server server_name  
-templateName templateName  
-validFrom validFromString  
[-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f deleteInvalidProcessTemplate.py  
-server server_name  
-node nodeName  
-templateName templateName  
-validFrom validFromString  
[-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f deleteInvalidProcessTemplate.py  
-cluster cluster_name  
-templateName templateName  
-validFrom validFromString  
[-profileName profileName]
```

其中：

-cluster *cluster_name*

集群的名称。如果业务流程编排器被配置在 WebSphere 集群中，那么此选项是必需的。可以指定集群名，也可以指定服务器名和节点名。

-node *nodeName*

在指定服务器名时，此选项是可选的。此名称用于标识节点。缺省值是本地节点。可以指定服务器名和节点名，也可以指定集群名。

-server *server_name*

服务器的名称。如果未指定集群名，那么此选项是必需的。可以指定服务器名和节点名，也可以指定集群名。

-templateName *templateName*

要删除的流程模板或任务模板的名称。

-validFrom *validFromString*

管理控制台中显示的模板的生效日期（UTC 格式）。此字符串的格式必须为：“yyyy-MM-ddThh:mm:ss”（年，月，日，T，小时，分钟，秒）。例如，2005-01-31T13:40:50。

-profileName *profileName*

用户定义的概要文件的名称。如果未使用缺省概要文件，那么应该指定此选项。

下一步做什么？

注：不推荐使用 JACL 版本的清理未用流程脚本 `deleteInvalidProcessTemplate.jacl`。此脚本在 `ProcessChoreographer` 目录的 `util` 子目录中，参数与上述参数相同，但不能指定 `-lang jython` 选项。

删除无效的人员任务模板

使用管理命令从业务流程编排器数据库中删除不再有效的人员任务模板。

开始之前

在开始执行本过程之前，要删除的模板所在的应用程序服务器必须处于运行状态。即，由于需要使用服务器连接，因此不能使用 `wsadmin` 的 `-conntype none` 选项。运行此命令并不需要任何特权，即使 WebSphere 管理安全性处于启用状态亦如此。如果业务流程编排器被配置在集群中，那么必须至少有一个集群成员处于运行状态。

关于此任务

对于数据库中那些未包含在 WebSphere 配置库中任何对应有应用程序中的模板，可以使用 `deleteInvalidTaskTemplate.py` 脚本将它们以及所有属于它们的对象除去。如果用户取消了应用程序的安装版本或者未将其存储在配置库中，那么会发生需要除去这些模板及其对象的情况。这些模板通常不起作用。它们不会显示在业务流程编排器资源管理器中。

在一些很罕见的情况下，无法对这些模板进行过滤。在这些情况下，必须使用下列脚本从数据库中除去这些模板。

不能使用这些脚本从数据库中除去有效应用程序的模板。将对此情况进行检查，如果相应的应用程序有效，那么将抛出 `ConfigurationError` 异常。

过程

1. 切换到管理脚本所在的业务流程编排器子目录。

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 和 i5/OS 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/admin
```

2. 从数据库中删除不再有效的人员任务模板。

在 Windows 系统上，要删除不再有效的人员任务模板，请输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root\bin\wsadmin -lang jython -f deleteInvalidTaskTemplate.py
    -server server_name
    -templateName templateName
    -validFrom validFromString
    -nameSpace nameSpace
    [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f deleteInvalidTaskTemplate.py
    -server server_name
    -node nodeName
    -templateName templateName
    -validFrom validFromString
    -nameSpace nameSpace
    [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f deleteInvalidTaskTemplate.py
    -cluster cluster_name
    -templateName templateName
```

```
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

在 UNIX 和 Linux 系统上，要删除不再有效的人员任务模板，请输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidTaskTemplate.py  
-server server_name  
-templateName templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidTaskTemplate.py  
-server server_name  
-node nodeName  
-templateName templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f deleteInvalidTaskTemplate.py  
-cluster cluster_name  
-templateName templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

在 i5/OS 系统上，要删除不再有效的人员任务模板，请输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin -lang jython -f deleteInvalidTaskTemplate.py  
-server server_name  
-templateName templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f deleteInvalidTaskTemplate.py  
-server server_name  
-node nodeName  
-templateName templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f deleteInvalidTaskTemplate.py  
-cluster cluster_name  
-templateName templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

其中：

-cluster *cluster_name*

集群的名称。如果业务流程编排器被配置在 WebSphere 集群中，那么此选项是必需的。可以指定集群名，也可以指定服务器名和节点名。

-node *nodeName*

在指定服务器名时，此选项是可选的。此名称用于标识节点。缺省值是本地节点。可以指定服务器名和节点名，也可以指定集群名。

-server *server_name*

服务器的名称。如果未指定集群名，那么此选项是必需的。可以指定服务器名和节点名，也可以指定集群名。

-templateName *templateName*

要删除的流程模板或任务模板的名称。

-validFrom *validFromString*

管理控制台中显示的模板的生效日期（UTC 格式）。此字符串的格式必须为：“yyyy-MM-ddThh:mm:ss”（年，月，日，T，小时，分钟，秒）。例如，2005-01-31T13:40:50。

-nameSpace *nameSpace*

任务模板的目标名称空间。

-profileName *profileName*

用户定义的概要文件的名称。如果未使用缺省概要文件，那么应该指定此选项。

下一步做什么？

注：不推荐使用 `JACL` 版本的清理未用人员查询脚本 `deleteInvalidTaskTemplate.jacl`。此脚本在 `ProcessChoreographer` 目录的 `util` 子目录中，参数与上述参数相同，但不能指定 `-lang jython` 选项。

删除已完成的流程实例

使用管理命令从业务流程编排器数据库中有选择地删除已达到“已完成”、“已终止”或“已失败”等结束状态的顶层流程实例。

开始之前

在开始执行本过程之前，要删除的流程实例所在的应用程序服务器必须处于运行状态。即，由于需要使用服务器连接，因此不能使用 `wsadmin` 的 `-conntype none` 选项。运行此命令并不需要任何特权，即使 `WebSphere` 管理安全性处于启用状态亦如此。如果业务流程编排器被配置在集群中，那么必须至少有一个集群成员处于运行状态。

关于此任务

如果顶层流程实例处于下列其中一种结束状态，那么认为此流程实例已完成：已完成、已终止、已结束或已失败。您可以指定条件，以便从数据库中有选择地删除顶层流程实例及其相关数据（例如活动实例、子流程实例和直接插入任务实例）。

过程

1. 切换到管理脚本所在的业务流程编排器子目录。

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 和 i5/OS 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/admin
```

2. 从数据库中删除流程实例。

在 Windows 系统上, 输入以下命令:

```
install_root\bin\wsadmin -lang jython -f deleteCompletedProcessInstances.py
  [[(-node nodeName) -server server_name) | (-cluster cluster_name)]
  (-all | -finished | -terminated | -failed )
  [-templateName templateName [-validFrom timestamp]]
  [-startedBy userID ]
  [-completedBefore timestamp]
  [-profileName profileName]
```

在 Linux 和 UNIX 系统上, 输入以下命令:

```
install_root/bin/wsadmin.sh -lang jython -f deleteCompletedProcessInstances.py
  [[(-node nodeName) -server server_name) | (-cluster cluster_name)]
  (-all | -finished | -terminated | -failed )
  [-templateName templateName [-validFrom timestamp]]
  [-startedBy userID ]
  [-completedBefore timestamp]
  [-profileName profileName]
```

在 i5/OS 系统上, 输入以下命令:

```
install_root/bin/wsadmin -lang jython -f deleteCompletedProcessInstances.py
  [[(-node nodeName) -server server_name) | (-cluster cluster_name)]
  (-all | -finished | -terminated | -failed )
  [-templateName templateName [-validFrom timestamp]]
  [-startedBy userID ]
  [-completedBefore timestamp]
  [-profileName profileName]
```

其中:

-node *nodeName*

在指定服务器名时, 此选项是可选的。此名称用于标识节点。缺省值是本地节点。可以指定服务器名和节点名, 也可以指定集群名。

-server *server_name*

服务器的名称。如果未指定集群名, 那么此选项是必需的。可以指定服务器名和节点名, 也可以指定集群名。

-cluster *cluster_name*

集群的名称。如果业务流程编排器被配置在 WebSphere 集群中, 那么此选项是必需的。可以指定集群名, 也可以指定服务器名和节点名。

-all|-finished|-terminated|-failed

根据状态指定要删除的流程实例。可以指定 finished、terminated、failed 与 all 的任意组合。

-templateName *templateName*

(可选) 指定要删除的流程模板的名称。如果指定了此选项, 那么还可以指定 validFrom。

-validFrom *timestamp*

管理控制台中显示的模板的生效日期 (UTC 格式)。此选项只能与 templateName 选项配合使用。timestamp 字符串的格式为: “yyyy-MM-dd[Thh:mm:ss]” (年, 月, 日, T, 小时, 分钟, 秒)。例如, 2006-11-20T12:00:00。如果仅指定了年月日, 那么小时、分钟和秒设置为 00:00:00。

-startedBy *userID*

(可选) 仅删除由给定用户标识启动的已完成流程实例。

-completedBefore *timestamp*

(可选) 删除给定时间前完成的已完成流程实例。 *timestamp* 字符串的格式为: “yyyy-MM-ddThh:mm:ss” (年, 月, 日, T, 小时, 分钟, 秒)。例如, 2006-07-20T12:00:00。

-profileName *profileName*

用户定义的概要文件的名称。如果未使用缺省概要文件, 那么应该指定此选项。

例如, 要删除节点 *myNode* 上服务器 *myServer* 中运行的所有由用户 *Antje* 启动并处于 *finished* 状态的流程实例, 请运行以下命令:

```
wsadmin -lang jython -f deleteCompletedProcessInstances.py
        -node myNode -server myServer
        -finished
        -startedBy Antje
```

结果

已从数据库中删除已完成的流程实例。

从观察器数据库中删除数据

通过使用管理命令, 可以从业务流程编排器观察器数据库中有选择地删除所有与所指定条件匹配的流程实例的数据。通过删除不必要的数据库, 可以提高生成报告时的性能。

关于此任务

可以通过三种方法删除流程实例的观察器信息:

- 要删除在指定时间前进入结束状态 *deleted* 的流程实例的观察器数据, 必须提供下列参数: **-deletedBefore** *timestamp*。
- 要删除特定模板版本的流程实例的观察器数据, 而不考虑其当前状态, 必须提供下列参数: **-templateName** *templateName* **-validFrom** *timestamp*。
- 要删除与特定模板版本相关并在所指定时间前进入所指定状态的流程实例的观察器数据, 必须提供下列参数: **-force** **-templateName** *template_name* **-validFrom** *timestamp* **-state** *state* **-reachedBefore** *timestamp*, 其中 **-templateName** *template_name* 和 **-validFrom** *timestamp* 是可选的。

要使用任何一种方法, 请执行下列操作:

过程

1. 确保应用程序服务器处于运行状态。如果业务流程编排器被配置在集群中, 那么必须至少有一个集群成员处于运行状态。
2. 切换到管理脚本所在的业务流程编排器子目录。

在 Windows 平台上, 输入以下命令:

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 和 i5/OS 平台上, 输入以下命令:

```
cd install_root/ProcessChoreographer/admin
```

3. 输入用于从数据库中删除特定流程实例的观察器数据的命令。

在 Windows 平台上, 输入以下命令:

```
install_root\bin\wsadmin -lang jython
-f observerDeleteProcessInstanceData.py
[ -user user_ID -password password ]
( [-node node_name] -server server_name ) | (-cluster cluster_name )
[ -profileName profile_name ]
[ -dataSource dataSource_JNDI_name ]
[ -dbSchemaName dbSchemaName ]
(
  -deletedBefore timestamp
  | ( -templateName template_name -validFrom timestamp )
  | ( -force [-templateName template_name -validFrom timestamp]
      -state state -reachedBefore timestamp )
)
```

在 Linux 和 UNIX 平台上, 输入以下命令:

```
install_root/bin/wsadmin.sh -lang jython
-f observerDeleteProcessInstanceData.py
[ -user user_ID -password password ]
( [-node node_name] -server server_name ) | (-cluster cluster_name )
[ -profileName profile_name ]
[ -dataSource dataSource_JNDI_name ]
[ -dbSchemaName dbSchemaName ]
(
  -deletedBefore timestamp
  | ( -templateName template_name -validFrom timestamp )
  | ( -force [-templateName template_name -validFrom timestamp]
      -state state -reachedBefore timestamp )
)
```

在 i5/OS 平台上, 输入以下命令:

```
install_root/bin/wsadmin -lang jython
-f observerDeleteProcessInstanceData.py
[ -user user_ID -password password ]
( [-node node_name] -server server_name ) | (-cluster cluster_name )
[ -profileName profile_name ]
[ -dataSource dataSource_JNDI_name ]
[ -dbSchemaName dbSchemaName ]
(
  -deletedBefore timestamp
  | ( -templateName template_name -validFrom timestamp )
  | ( -force [-templateName template_name -validFrom timestamp]
      -state state -reachedBefore timestamp )
)
```

输入以下命令:

```
install_root/bin/wsadmin.sh -lang jython
-f observerDeleteProcessInstanceData.py
[ -user user_ID -password password ]
( [-node node_name] -server server_name ) | (-cluster cluster_name )
[ -profileName profile_name ]
[ -dataSource dataSource_JNDI_name ]
[ -dbSchemaName dbSchemaName ]
(
  -deletedBefore timestamp
  | ( -templateName template_name -validFrom timestamp )
  | ( -force [-templateName template_name -validFrom timestamp]
      -state state -reachedBefore timestamp )
)
```

其中:

-user *user_ID* **-password** *password*

如果已对服务器启用 WebSphere 管理安全性，那么必须提供有权从数据库中删除数据的用户标识以及相关密码。

-node *node_name*

此名称用于标识节点。此参数是可选的。缺省值是本地节点。

-server *server_name*

服务器的名称。缺省值是缺省服务器。如果指定此参数，那么不能指定 `cluster` 参数。

-cluster *cluster_name*

集群的名称。如果指定此参数，那么不能指定 `server` 参数。

-profileName *profile_name*

用户定义的 WebSphere 概要文件的名称。如果未使用缺省概要文件，那么应该指定此选项。

-dataSource *datasource_JNDI_name*

因为服务器或集群可以有多个观察器数据库，所以此参数指定此命令要处理的数据库。缺省值是 `jdbc/BPEDB`。

-dbSchemaName *dbSchemaName*

如果对观察器数据库设置了特定模式名，那么请使用此参数。

-deletedBefore *timestamp*

删除在指定时间前进入“已删除”状态的流程实例的所有观察器数据。

timestamp

日期和时间以全球标准时间（UTC）表示，格式如下：`yyyy-MM-dd[Thh:mm:ss]`（年，月，日，T，小时，分钟，秒）。例如，`2008-07-20T12:00:00`。如果仅指定了年月日，那么小时、分钟和秒设置为 `00:00:00`。

-templateName *template_name*

删除属于所指定模板版本的实例的所有观察器数据。

-validFrom *timestamp*

如果指定了 `templateName` 选项，那么必须指定此选项。

timestamp

日期和时间以全球标准时间（UTC）表示，格式如下：`yyyy-MM-ddThh:mm:ss`（年，月，日，T，小时，分钟，秒）。例如，`2008-07-20T12:00:00`。

-force

强制删除与所有模板或所指定模板版本相关并在给定时间前进入指定状态的流程实例的所有观察器数据。如果使用此选项，那么还必须指定 `-state` 和 `-reachedBefore` 选项。`-templateName` 和 `-validFrom` 选项是可选的。

-state *state*

指定下列其中一种状态：`running`、`terminated`、`suspended`、`failed`、`finished` 或 `compensated`。

-reachedBefore *timestamp*

指定进入所指定状态的时间。

timestamp

日期和时间以全球标准时间（UTC）表示，格式如下：`yyyy-MM-`

`dd[Thh:mm:ss]` (年, 月, 日, T, 小时, 分钟, 秒)。例如, 2008-07-20T12:00:00。如果仅指定了年月日, 那么小时、分钟和秒设置为00:00:00。

例如, 要删除 2007 年 1 月 2 日起生效的流程模板 `my_template` 的实例的所有观察器数据, 这些实例在 2007 年 7 月 20 日中午前启动并在服务器 `my_server` 中节点 `my_node` 上运行, 请运行以下命令:

```
wsadmin -lang jython -f observerDeleteProcessInstanceData.py
        -node my_node -server my_server
        -force -templateName my_template -validFrom 2007-01-02T12:00:00
        -state running -reachedBefore 2007-07-20T12:00:00
```

结果

如果成功, 此工具将报告删除了观察器数据的实例数以及从数据库中删除的表条目数。否则, 将报告错误信息, 并且不会对数据库进行更改。

使用管理命令查询和重放已失败的消息

使用管理命令来确定是否存在业务流程或人员任务的失败消息并再次尝试处理那些消息。

开始之前

在开始执行本过程之前, 必须符合下列条件:

- 所要查询或重放的消息所在的应用程序服务器必须处于运行状态。即, 由于需要使用服务器连接, 因此不能使用 `wsadmin` 的 `-conntype none` 选项。
- 如果业务流程编排器被配置在集群中, 那么必须至少有一个集群成员处于运行状态。
- 如果已启用 WebSphere 管理安全性, 那么您必须具有操作员权限。

关于此任务

如果处理内部消息时发生了问题, 那么此消息最终将进入保留队列或挂起队列。要确定是否存在任何已失败的消息以及如何将那些消息再次发送至内部队列:

过程

1. 切换到管理脚本所在的业务流程编排器子目录。

在 Windows 平台上, 输入以下命令:

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 和 i5/OS 平台上, 输入以下命令:

```
cd install_root/ProcessChoreographer/admin
```

2. 查询保留队列和挂起队列中的失败消息数。

在 Windows 系统上, 输入下列其中一个命令。这些命令之间的差别已突出显示:

```
install_root\bin\wsadmin -lang jython -f queryNumberOfFailedMessages.py
                        -cluster cluster_name
                        [ -bfm | -htm ]
                        [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f queryNumberOfFailedMessages.py
                        -node nodeName
                        -server server_name
                        [ -bfm | -htm ]
                        [-profileName profileName]
```

在 Linux 和 UNIX 系统上, 输入下列其中一个命令。这些命令之间的差别已突出显示:

```
install_root/bin/wsadmin.sh -lang jython -f queryNumberOfFailedMessages.py
                             -cluster cluster_name
                             [ -bfm | -htm ]
                             [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f queryNumberOfFailedMessages.py
                             -node nodeName
                             -server server_name
                             [ -bfm | -htm ]
                             [-profileName profileName]
```

在 i5/OS 系统上, 输入下列其中一个命令。这些命令之间的差别已突出显示:

```
install_root/bin/wsadmin -lang jython -f queryNumberOfFailedMessages.py
                          -cluster cluster_name
                          [ -bfm | -htm ]
                          [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f queryNumberOfFailedMessages.py
                          -node nodeName
                          -server server_name
                          [ -bfm | -htm ]
                          [-profileName profileName]
```

其中:

-cluster *cluster_name*

集群的名称。如果业务流程编排器被配置在 WebSphere 集群中, 那么此选项是必需的。

-node *nodeName*

在指定服务器名时, 此选项是可选的。此名称用于标识节点。缺省值是本地节点。

-server *server_name*

服务器的名称。如果未指定集群名, 那么此选项是必需的。

-bfm|-htm

这些关键字是可选的。当这两个选项都未指定时, 缺省情况是同时显示业务流程和人员任务的所有失败消息。如果只想显示业务流程管理器的挂起队列和保留队列中的消息数, 请指定 **-bfm** 选项。如果只想显示人员任务管理器挂起队列中的消息数, 请指定 **-htm** 选项。

-profileName *profileName*

用户定义的概要文件的名称。如果未使用缺省概要文件, 那么应该指定此选项。

如果要检查本地节点上的服务器, 请输入:

```
wsadmin -lang jython -f queryNumberOfFailedMessages.py -server server_name
```

3. 重放挂起队列和/或保留队列中的所有失败消息。

在 Windows 系统上，输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root\bin\wsadmin -lang jython -f replayFailedMessages.py
    -cluster cluster_name
    -queue replayQueue
    [ -bfm | -htm ]
    [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f replayFailedMessages.py
    -node nodeName
    -server server_name
    -queue replayQueue
    [ -bfm | -htm ]
    [-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f replayFailedMessages.py
    -server server_name
    -queue replayQueue
    [ -bfm | -htm ]
    [-profileName profileName]
```

在 Linux 和 UNIX 系统上，输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin.sh -lang jython -f replayFailedMessages.py
    -cluster cluster_name
    -queue replayQueue
    [ -bfm | -htm ]
    [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f replayFailedMessages.py
    -node nodeName
    -server server_name
    -queue replayQueue
    [ -bfm | -htm ]
    [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f replayFailedMessages.py
    -server server_name
    -queue replayQueue
    [ -bfm | -htm ]
    [-profileName profileName]
```

在 iOS 系统上，输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin -lang jython -f replayFailedMessages.py
    -cluster cluster_name
    -queue replayQueue
    [ -bfm | -htm ]
    [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f replayFailedMessages.py
    -node nodeName
    -server server_name
    -queue replayQueue
    [ -bfm | -htm ]
    [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f replayFailedMessages.py
    -server server_name
    -queue replayQueue
    [ -bfm | -htm ]
    [-profileName profileName]
```

其中：

-queue *replayQueue*

(可选) 指定要重放的队列。 *replayQueue* 可以具有下列其中一个值:

holdQueue (这是缺省值)

retentionQueue (仅当指定了 `-bfm` 选项时才有效)

both (如果指定了 `-bfm` 选项, 那么此值无效)

-cluster *cluster_name*

集群的名称。如果业务流程编排器被配置在 WebSphere 集群中, 那么此选项是必需的。

-node *nodeName*

在指定服务器名时, 此选项是可选的。此名称用于标识节点。缺省值是本地节点。

-server *server_name*

服务器的名称。如果未指定集群名, 那么此选项是必需的。

-bfml-htm

这些关键字是可选的, 并且互斥。当这两个选项都未指定时, 缺省情况是同时重放业务流程和人员任务的失败消息。如果只想重放业务流程的消息, 请指定 `-bfm` 选项。如果只想重放人员任务的消息, 请指定 `-htm` 选项。

-profileName *profileName*

用户定义的概要文件的名称。如果未使用缺省概要文件, 那么应该指定此选项。

下一步做什么?

注: 不推荐使用 JACL 版本的清理未用人员查询脚本 `replayFailedMessages.jacl`。此脚本在 `ProcessChoreographer` 目录的 `util` 子目录中, 参数与上述参数相同, 但不能指定 `-lang jython` 选项。

相关概念

第 26 页的『从基础结构故障恢复』

业务流程管理器提供了用于处理基础结构临时故障的功能。

使用管理命令刷新人员查询结果

人员查询的结果是静态的。您可以使用管理命令来刷新人员查询。

开始之前

在开始执行本过程之前, 必须符合下列条件:

- 所要查询或重放的消息所在的应用程序服务器必须处于运行状态。即, 由于需要使用服务器连接, 因此不能使用 `wsadmin` 的 `-conntype none` 选项。
- 如果业务流程编排器被配置在集群中, 那么必须至少有一个集群成员处于运行状态。
- 如果已启用 WebSphere 管理安全性, 那么您必须具有操作员权限。

关于此任务

业务流程编排器将根据人员目录（例如轻量级目录访问协议（LDAP）服务器）进行求值的人员查询的结果高速缓存在运行时数据库中。如果人员目录发生更改，那么可以强制再次对人员分配进行评估。

过程

1. 切换到管理脚本所在的业务流程编排器子目录。

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 和 i5/OS 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/admin
```

2. 强制再次对人员分配进行评估。

在 Windows 平台上，输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root\bin\wsadmin -lang jython -f refreshStaffQuery.py  
-server server_name  
[-processTemplate templateName |  
(-taskTemplate templateName [-nameSpace nameSpace]) |  
-userlist username{,username}...]  
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f refreshStaffQuery.py  
-node nodeName  
-server server_name  
[-processTemplate templateName |  
(-taskTemplate templateName [-nameSpace nameSpace]) |  
-userlist username{,username}...]  
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f refreshStaffQuery.py  
-cluster cluster_name  
[-processTemplate templateName |  
(-taskTemplate templateName [-nameSpace nameSpace]) |  
-userlist username{,username}...]  
[-profileName profileName]
```

在 Linux 和 UNIX 平台上，输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin.sh -lang jython -f refreshStaffQuery.py  
-server server_name  
[-processTemplate templateName |  
(-taskTemplate templateName [-nameSpace nameSpace]) |  
-userlist username{,username}...]  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f refreshStaffQuery.py  
-node nodeName  
-server server_name  
[-processTemplate templateName |  
(-taskTemplate templateName [-nameSpace nameSpace]) |  
-userlist username{,username}...]  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f refreshStaffQuery.py  
-cluster cluster_name  
[-processTemplate templateName |  
(-taskTemplate templateName [-nameSpace nameSpace]) |  
-userlist username{,username}...]  
[-profileName profileName]
```

在 i5/OS 平台上，输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin -lang jython -f refreshStaffQuery.py
  -server server_name
  [-processTemplate templateName |
  (-taskTemplate templateName [-nameSpace nameSpace]) |
  -userlist username{,username}...]
  [-profileName profileName]

install_root/bin/wsadmin -lang jython -f refreshStaffQuery.py
  -node nodeName
  -server server_name
  [-processTemplate templateName |
  (-taskTemplate templateName [-nameSpace nameSpace]) |
  -userlist username{,username}...]
  [-profileName profileName]

install_root/bin/wsadmin -lang jython -f refreshStaffQuery.py
  -cluster cluster_name
  [-processTemplate templateName |
  (-taskTemplate templateName [-nameSpace nameSpace]) |
  -userlist username{,username}...]
  [-profileName profileName]
```

其中：

-cluster *cluster_name*

集群的名称。如果业务流程编排器被配置在 WebSphere 集群中，那么此选项是必需的。

-node *nodeName*

在指定服务器名时，此选项是可选的。此名称用于标识节点。缺省值是本地节点。

-server *server_name*

服务器的名称。如果未指定集群名，那么此选项是必需的。

-processTemplate *templateName*

流程模板的名称。将刷新属于此流程模板的人员分配。

-taskTemplate *templateName*

任务模板的名称。将刷新属于此任务模板的人员分配。

-nameSpace *nameSpace*

任务模板的名称空间。

-userlist *userName*

以逗号分隔的用户名列表。将刷新包含所指定名称的人员分配。可以将用户列表括在引号中。如果省略引号，那么用户列表的用户名之间不能有空格。

-profileName *profileName*

用户定义的概要文件的名称。如果未使用缺省概要文件，那么应该指定此选项。

注：如果既未指定 *templateName* 也未指定 *userlist*，那么将刷新数据库中存储的所有人员查询。为了提高性能，应该避免执行此操作。

注：不推荐使用 JACL 版本的刷新人员查询脚本 `refreshStaffQuery.jacl`。此脚本在 `ProcessChoreographer` 目录的 `util` 子目录中，参数与上述参数相同，但不能指定 `-lang jython` 选项。

相关概念

第 76 页的『管理人员分配条件和人员解析结果』

与任务授权角色相关联的人员分配条件在所部署任务模板或任务实例的整个生存期内有效。

使用管理命令除去未使用的人员查询结果

使用管理命令从数据库中除去未使用的人员查询结果。

开始之前

在开始执行本过程之前，必须符合下列条件：

- 用于删除未使用的人员查询的应用程序服务器必须处于运行状态。即，由于需要使用服务器连接，因此不能使用 `wsadmin` 的 `-conntype none` 选项。
- 如果业务流程编排器被配置在集群中，那么必须至少有一个集群成员处于运行状态。
- 如果已启用 WebSphere 管理安全性，那么您必须具有操作员权限。

关于此任务

业务流程编排器在运行时数据库中维护已求值的人员查询的用户名列表。虽然使用了那些人员查询的流程实例和人员任务已完成，但仍然在数据库中维护用户名列表，直到相应的业务流程应用程序被卸载为止。

如果数据库的大小影响了性能，那么可以除去数据库表中高速缓存的未使用的人员列表。

过程

1. 切换到管理脚本所在的业务流程编排器子目录。

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 和 iOS 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/admin
```

2. 除去未使用的人员列表。

在 Windows 平台上，输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root\bin\wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py  
-server server_name  
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py  
-node nodeName  
-server server_name  
[-profileName profileName]
```

```
install_root\bin\wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py  
-cluster cluster_name  
[-profileName profileName]
```

在 Linux 和 UNIX 平台上，输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin.sh -lang jython -f cleanupUnusedStaffQueryInstances.py
    -server server_name
    [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f cleanupUnusedStaffQueryInstances.py
    -node nodeName
    -server server_name
    [-profileName profileName]
```

```
install_root/bin/wsadmin.sh -lang jython -f cleanupUnusedStaffQueryInstances.py
    -cluster cluster_name
    [-profileName profileName]
```

在 i5/OS 平台上，输入下列其中一个命令。这些命令之间的差别已突出显示：

```
install_root/bin/wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
    -server server_name
    [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
    -node nodeName
    -server server_name
    [-profileName profileName]
```

```
install_root/bin/wsadmin -lang jython -f cleanupUnusedStaffQueryInstances.py
    -cluster cluster_name
    [-profileName profileName]
```

其中：

-cluster *cluster_name*

集群的名称。如果业务流程编排器被配置在 WebSphere 集群中，那么此选项是必需的。

-node *nodeName*

在指定服务器名时，此选项是可选的。此名称用于标识节点。缺省值是本地节点。

-server *server_name*

服务器的名称。如果未指定集群名，那么此选项是必需的。

-profileName *profileName*

用户定义的概要文件的名称。如果未使用缺省概要文件，那么应该指定此选项。

结果

将显示从数据库中删除的条目数。

下一步做什么？

注：不推荐使用 `JACL` 版本的清理未用人员查询脚本 `cleanupUnusedStaffQueryInstances.jacl`。此脚本在 `ProcessChoreographer` 目录的 `util` 子目录中，参数与上述参数相同，但不能指定 `-lang jython` 选项。

第 7 章 业务流程编排器资源管理器入门

根据用户角色的不同，您可以使用业务流程编排器资源管理器来管理业务流程和人员任务，也可以处理分配给您的任务。

关于此任务

可以使用业务流程编排器资源管理器来执行下列任务：

- 如果您是业务管理员，那么可以管理业务流程的生命周期以及修复业务流程。例如，可以重新启动或强制完成单个的活动或者对整个业务流程进行补偿。如果补偿失败，那么可以重试、跳过或停止流程实例。此外，还可以添加和更新业务流程和活动的定制属性。
- 如果您是人员任务管理员，那么可以管理人员任务的生命周期以及工作分配。例如，可以对用户分配职责或者管理用户的不在场处理方式和替代者。还可以更改人员任务的优先级和业务类别以及添加或更新定制属性。
- 如果您是业务用户，那么可以使用业务流程编排器资源管理器来处理分配给您的任务。例如，可以启动业务流程、服务和人员任务，也可以处理、编辑、保存、完成或释放人员任务。此外，还可以标记您已离开以及定义替代者。

此外，业务流程编排器资源管理器提供了搜索功能，您可以通过此功能发现业务流程及其需要注意的相关活动和人员任务。例如，可以检查这些实例的状态，可以在相关实例和模板之间进行导航，并可以检索包含相关活动和人员任务的流程状态图形视图。

相关任务

第 313 页的『管理任务模板和任务实例』

使用管理控制台或管理命令来管理任务模板。使用业务流程编排器资源管理器来处理任务实例。

第 317 页的『管理工作分配』

启动任务后，可能需要管理该任务的工作分配，例如，以更好地在工作组成员之间分配工作量。

第 315 页的『创建和启动任务实例』

您可以根据有权使用的任务模板来创建和启动任务实例。

第 315 页的『处理任务』

要处理某个任务，必须先声明该任务，然后执行完成该任务所需的操作。

业务流程编排器资源管理器用户界面

业务流程编排器资源管理器是一个独立的 Web 应用程序，它提供了一组用于管理业务流程和人员任务的管理功能。业务流程编排器资源管理器用户界面由任务栏、导航窗格和工作空间组成。

下图说明业务流程编排器资源管理器用户界面的布局。



用户界面的主要区域如下所示。

任务栏

对于所有用户，任务栏提供了用于从业务流程编排器资源管理器注销、指定离开设置以及访问联机帮助选项。如果您拥有系统管理员权限，那么任务栏还包含下列选项：

定制 要对此业务流程编排器资源管理器实例的导航窗格添加视图和除去视图，请选择此选项。您还可以定义用户登录后看到的视图。

定义视图

要定义用户组的定制视图，请选择此选项。

定义替代者

选择此选项以定义用户的离开设置。

导航窗格

导航窗格包含可用于管理对象的视图链接，这些对象包括已启动的流程实例以及您有权管理的人员任务。缺省用户界面包含指向业务流程和任务的预定义视图链接。

通过对导航窗格添加和除去预定义视图以及定义要对导航窗格添加的定制视图，系统管理员可以对缺省导航窗格的内容进行定制。所有用户都可以根据导航窗格来定义个性化视图。

工作空间

工作空间包含一些页面，这些页面用于查看和管理与业务流程和人员任务相关的对象。您可以通过单击导航窗格中的链接、单击操作栏中的操作或单击工作空间页面中的链接来访问这些页面。

业务流程编排器资源管理器的导航窗格

使用导航窗格来访问用于管理业务流程和人员任务对象（例如流程实例和工作分配）的视图。缺省用户界面包含指向业务流程和任务的预定义视图链接。您还可以定义自己的个性化视图，这些视图将被添加到导航窗格。此外，如果您是系统管理员，那么可以定义可供所有用户使用的定制视图。

可用的操作

在导航窗格中，可以执行下列操作：

- 转到某个视图。

要转到某个视图，请单击该视图的名称。






- 折叠和展开组。

在导航窗格中，单击某个项旁边的箭头以展开或折叠该项。

- 定义新搜索。

单击**新建搜索**图标（）以搜索对象或定义个性化视图。

根据视图类型的不同，还可以使用弹出菜单执行其他操作。图标指示了弹出菜单是否可用。

- 要删除视图，请单击**删除**图标 。
- 要修改视图，请单击**修改**图标 。
- 要创建视图副本并修改该副本，请单击**复制**图标 。
- 要在列表中将视图上移或下移，请单击**上移**图标  或**下移**图标 。

导航窗格中的预定义视图

缺省导航窗格包含下列各组视图。根据系统管理员是否对业务流程编排器资源管理器的导航窗格添加或删除了视图，该导航窗格中显示的视图可能会有所不同。如果没有为视图组定义视图，那么不会显示该组。

流程模板

流程模板组包含以下视图：

我的流程模板

此视图显示流程模板列表。在此视图中，可以显示流程模板及其结构的信息、显示某个模板的相关流程实例列表以及启动流程实例。

流程实例

流程实例组包含下列视图:

由我启动

此视图显示您启动的流程实例。在此视图中，可以监视流程实例的进度以及列示与其相关的活动、流程或任务。

由我管理

此视图显示您有权管理的流程实例。在此视图中，可以对流程实例执行操作，例如暂挂和恢复流程以及监视流程实例中的活动进度。

关键流程

此视图显示处于运行状态的流程实例，这些流程实例包含处于“已停止”状态的活动。在此视图中，可以对流程实例执行操作，也可以列示活动并对它们执行操作。

已终止流程

此视图显示处于终止状态的流程实例。在此视图中，可以对这些流程实例执行操作。

失败的补偿

此视图显示微流程的已失败的补偿操作。

活动实例

缺省情况下，活动实例组未包含任何视图。因此，此组未显示在缺省导航窗格中。

任务模板

任务模板组包含以下视图:

我的任务模板

此视图显示任务模板列表。在此视图中，可以创建和启动任务实例以及显示模板的相关任务实例列表。

任务实例

任务实例组包含下列视图:

我的待执行任务

此视图显示您有权处理的任务实例列表。在此视图中，可以处理任务实例、释放已声明的任务实例或者将任务实例移交给另一用户。

全部任务

此视图显示所有以您作为所有者、潜在所有者或编辑者的任务。在此视图中，可以处理任务实例、释放已声明的任务实例或者将任务实例移交给另一用户。

由我启动

此视图显示您启动的任务实例。在此视图中，可以处理任务实例、释放已声明的任务实例或者将任务实例移交给另一用户。

由我管理

此视图显示您有权管理的任务实例。在此视图中，可以对任务实例执行操作，例如暂挂和恢复流程、创建任务实例的工作项以及显示任务实例的当前工作项列表。

我的升级内容

此视图显示登录用户的所有升级。

视图类型



导航窗格可以包含下列类型的视图。根据视图的不同，可以使用弹出菜单执行其他操作。

缺省导航窗格中的预定义视图。


仅当系统管理员未在定制导航树和登录视图页面中更改导航窗格时，这些视图组才可用。这些视图未提供弹出菜单。

系统管理员对导航窗格添加的定制视图和预定义视图。

业务用户可以通过单击视图名转到该视图。对于系统管理员而言，可以使用弹出菜单。

- 预定义视图由**预定义视图**图标  指示。系统管理员可以使用弹出菜单更改这些视图在导航窗格中的位置。
- 定制视图由**定制视图**图标  指示。系统管理员可以删除、编辑、复制和移动这些视图。

个性化视图。

这些视图由**定制视图**图标  指示。这些视图仅对创建这些视图的用户可视。该用户可以删除、编辑、复制和移动这些视图。

启动业务流程编排器资源管理器

业务流程编排器资源管理器是可以作为业务流程容器配置一部分安装的 Web 应用程序。在 Web 浏览器中使用业务流程编排器资源管理器之前，必须已安装业务流程容器、人员任务容器和业务流程编排器资源管理器应用程序，并且该应用程序必须处于运行状态。

关于此任务

要启动业务流程编排器资源管理器，请完成下列步骤。

过程

1. 使用 Web 浏览器打开业务流程编排器资源管理器 URL。

此 URL 具有以下格式。URL 值取决于您为安装版本配置的虚拟主机和上下文根。

`http://app_server_host:port_no/context_root`

其中：

app_server_host

应用程序服务器所在主机的网络名，此应用程序服务器提供了您所要使用的业务流程应用程序。

port_no

业务流程编排器资源管理器使用的端口号。随系统配置的不同，端口号也会有所变化。缺省端口号是 9080。

context_root

应用程序服务器上业务流程编排器资源管理器应用程序的根目录。缺省值是 bpc。

2. 如果已启用安全性，那么必须输入用户标识和密码，然后单击**登录**。

结果

将显示业务流程编排器资源管理器的第一个页面。缺省情况下，此页面显示了我的待执行任务视图。

定制业务流程编排器资源管理器

业务流程编排器资源管理器提供了通用的用户界面，管理员可以使用此界面来管理业务流程和人员任务，企业用户可以使用此界面来处理被分配的任务。由于这是通用的界面，因此，您可能希望为特定的业务流程编排器资源管理器实例定制此界面，以满足分配给此实例的用户组的业务需求。

关于此任务

可以通过各种方法对此用户界面进行定制。

为不同的用户组定制业务流程编排器资源管理器界面

缺省业务流程编排器资源管理器用户界面中的导航窗格包含一组预定义视图链接。我的待执行任务视图是用户登录后显示的缺省视图。如果您具有其中一个业务流程编排器系统管理员角色，那么可以对导航窗格中显示的链接、用户在登录后看到的视图以及各个视图中显示的信息进行定制。

开始之前

要定制界面，您必须拥有 `BPCSystemAdministrator` 权限。

关于此任务

例如，业务流程编排器资源管理器的缺省用户界面不包含用于处理业务状态机的视图。您可以添加预定义视图，以处理业务状态机的流程模板和流程实例。

此外，您可能想为负责处理客户订单的用户和负责处理客户服务查询的用户提供不同的界面。可以定制业务流程编排器资源管理器的实例，以使其符合那些被分配到该实例的用户的 workflows 模式。

要定制业务流程编排器资源管理器的缺省用户界面，请完成下列步骤。

过程

1. 定制导航窗格中包含的视图集以及缺省登录视图。
 - a. 单击任务栏中的**定制**。
 - b. 在定制导航树和登录视图页面中，选中要包括在导航窗格中的视图，并取消选中要从此窗格中除去的视图。
 - c. 选择用户登录到业务流程编排器资源管理器时看到的视图。

列表包含上一个步骤中选择的视图以及您在搜索和定义定制视图页面中创建的任何定制视图（请参阅步骤 第 291 页的 2）。
 - d. 要保存更改，请单击**保存**。

要将此实例的视图恢复为缺省视图，请单击**恢复缺省设置**。此操作将使导航窗格恢复为包含预定义视图。导航窗格中的定制视图不受此操作影响。

2. 定制视图。

您可以指定此业务流程编排器资源管理器实例的各个视图中显示的信息。

- a. 单击任务栏中的**定义视图**。
- b. 在搜索和定义定制视图页面中，选择要定制的视图类型，例如“流程模板”。
- c. 在视图的搜索和定义定制视图页面中，指定搜索条件。
- d. 使用视图属性选项卡来选择要包括在视图中的属性以及指定列表属性。

如果这是任务实例视图或流程实例视图，那么请单击**视图设置**以选择一组要添加至视图操作栏的操作，并可以指定对于系统管理员和系统监视员要包括在视图中的项。

- 选择视图类型:
 - 要将管理操作添加至视图操作栏，请选择**管理实例**。
 - 要在操作栏中添加一组操作以使登录用户能够处理实例，请选择**使用实例**。
- 对于系统管理员和系统监视员而言，可以对搜索结果进行限制，以便只显示他们自己的实例的结果:
 - 要在视图中显示所有与搜索条件匹配的项，请选中**全部实例**。这将显示所有的项，而不考虑此系统管理员是否有这些项的工作项。
 - 要仅显示登录用户所拥有的工作项的对应项，请选中**个人实例**。
- e. 在**视图名称**字段中输入该视图的显示名，然后单击**保存**。

新视图将显示在导航窗格中。用户下次登录到业务流程编排器资源管理器时将看到这个新视图。

为业务状态机的流程模板定义视图

虽然产品提供了业务状态机的流程模板的预定义视图，但您可能想为此类模板定义自己的视图。

开始之前

要创建定制视图，您必须拥有 `BPCSystemAdministrator` 权限。

过程

1. 单击任务栏中的**定义视图**。
2. 在搜索并定义视图页面中，选择**搜索流程模板和定义定制视图**。
3. 单击**属性过滤器** → **定制属性过滤器**。
 - a. 添加一个定制属性并指定下列设置:
 - 在**属性名字段**中，输入 `generatedBy`。
 - 在**属性值字段**中，输入 `BusinessStateMachine`。
 - b. 单击**添加**。
 - c. 根据需要，添加其他定制属性。
4. 单击**视图属性** → **列表列**。
 - a. 在“定制属性的列表列”中，添加一个定制属性并指定下列设置:
 - 在**属性名字段**中，输入 `generatedBy`。

- 在**显示名字**字段中，输入该列的显示名，然后单击**添加**。
 - b. 对所选列的列表添加其他列或删除列。
5. 在**视图名称**字段中输入该查询的显示名，然后单击**保存**。

结果

缺省情况下，将在导航窗格的“流程模板”组中添加新视图的链接。用户下次登录到业务流程编排器资源管理器时将看到此视图。

为业务状态机的流程实例定义视图

虽然产品提供了业务状态机的流程实例的预定义视图，但您可能想为此类流程实例定义自己的视图。

开始之前

要创建定制视图，您必须拥有 `BPCSystemAdministrator` 权限。

过程

1. 单击任务栏中的**定义视图**。
2. 在搜索并定义视图页面中，选择**搜索流程实例和定义定制视图**。
3. 单击**属性过滤器** → **定制属性过滤器**。
 - a. 添加一个定制属性并指定下列设置：
 - 在**属性名字**字段中，输入 `generatedBy`。
 - 在**属性值**字段中，输入 `BusinessStateMachine`。
 - b. 单击**添加**。
 - c. 根据需要，添加其他定制属性。
4. 单击**视图属性** → **列表列**。
 - a. 在“查询属性的列表列”中，添加下列查询属性。
 - 要在视图中添加业务状态信息，请在**属性名字**字段中输入 `name`，在**变量名字**字段中输入 `DisplayState`，并在**名称空间**字段中输入 `tns`，其中 `tns` 是业务状态机的目标名称空间并带有 `-process` 后缀。并且，请在**显示名字**字段中指定该列的显示名，然后单击**添加**。
 - 要在视图中添加关联信息，请在**属性名字**字段、**变量名字**字段和**名称空间**字段中提供适当的信息。这些值从业务状态机的定义派生。并且，请在**显示名字**字段中提供该列的显示名。

属性名 为业务状态机定义的关联属性的名称。

变量名 如果关联集由输入参数初始化，那么变量名的格式为：

`operation_name_Input_operation_parameter_name`

其中 `operation_name` 是用于脱离初始状态的操作的名称。

如果关联集由输出参数启动，那么变量名的格式为：

`operation_name_Output_operation_parameter_name`

名称空间

查询属性的名称空间，其中 `tns` 是业务状态机的目标名称空间并带有 `-process` 后缀。

- b. 添加其他定制属性或查询属性，或者对所选列的列表添加列或删除列。
5. 在**视图名称**字段中输入查询名，然后单击**保存**。

结果

缺省情况下，将在导航窗格的“流程实例”组中添加新视图的链接。用户下次登录到业务流程编排器资源管理器时将看到此视图。


对业务流程编排器资源管理器界面进行个性化定制

缺省业务流程编排器资源管理器用户界面中的导航窗格包含一组链接，这些链接指向预定义视图和系统管理员定义的视图。例如，您可以对导航窗格添加自己的视图以监视特定任务或流程的进度。

关于此任务

在业务流程编排器资源管理器中，请完成下列步骤以对用户界面进行个性化定制。

过程

1. 在要定义新视图的导航树部分中，单击**新建搜索**图标 ()。
2. 在视图的搜索和定义个性化视图页面中，指定搜索条件。
3. 使用视图属性选项卡来选择要包括在视图中的属性以及指定列表属性。

如果这是任务实例视图或流程实例视图，那么请单击**视图设置**以选择一组要添加至视图操作栏的操作。如果您是系统管理员或系统监视员，那么请指定该视图中包括的项。

- 选择视图类型:
 - 要将管理操作添加至视图操作栏，请选择**管理实例**。
 - 要在操作栏中添加一组操作以使登录用户能够处理实例，请选择**使用实例**。
 - 如果您是系统管理员或系统监视员，那么可以对搜索结果进行限制，以便只显示您自己的实例的结果:
 - 要在视图中显示所有与搜索条件匹配的项，请选中**全部实例**。这将显示所有的项，而不考虑此系统管理员是否有这些项的工作项。
 - 要仅显示登录用户所拥有的工作项的对应项，请选中**个人实例**。
4. 在**视图名称**字段中输入该视图的显示名，然后单击**保存**。

结果

新视图将显示在导航窗格中。

更改缺省 Web 应用程序的外观

业务流程编排器资源管理器提供了简单易用的 Web 用户界面，此界面基于 JavaServer Pages (JSP) 文件和 JavaServer Faces (JSF) 组件。级联样式表 (CSS) 控制 Web 界面的显示方式。您可以修改样式表，以便将此用户界面修改为与特定的外观一致，而不必编写任何新代码。

开始之前

要修改样式表，您必须深入了解级联样式表。

关于此任务

例如，您可以更改 CSS，以使缺省界面符合企业身份准则。

过程

修改样式表。缺省样式表 `style.css` 包含页眉、导航窗格和内容窗格中各个元素的样式。

相关概念

第 285 页的『业务流程编排器资源管理器用户界面』

业务流程编排器资源管理器是一个独立的 Web 应用程序，它提供了一组用于管理业务流程和人员任务的管理功能。业务流程编排器资源管理器用户界面由任务栏、导航窗格和工作空间组成。

业务流程编排器资源管理器界面中使用的样式

`style.css` 文件包含样式。您可以对这些样式进行更改，以修改缺省用户界面的外观。

`style.css` 文件包含缺省用户界面中下列元素的样式：

- 『条幅』
- 第 295 页的『页脚』
- 第 295 页的『菜单栏』
- 第 295 页的『“登录”页面』
- 第 295 页的『导航器』
- 第 295 页的『内容面板』
- 第 296 页的『命令栏』
- 第 296 页的『列表』
- 第 296 页的『详细信息面板』
- 第 296 页的『消息数据』
- 第 296 页的『选项卡式窗格』
- 第 297 页的『搜索页面』
- 第 297 页的『错误详细信息』

此文件在以下目录中：

```
<profile_root>\installedApps\<node_name>\<explorer_instance>\bpcexplorer.war\theme
```

条幅

样式名	描述
<code>.banner</code>	条幅的分割线。
<code>.banner_left</code>	条幅中的一个分割区域。此分割区域用于嵌入应用程序的标题图像。
<code>.banner_right</code>	条幅中的一个分割区域。例如，可以使用此分割区域来显示其他徽标。

页脚

样式名	描述
.footer	页脚的分割线。
.footer_left	页脚中的一个分割区域，例如，您可以使用此分割区域来显示应用程序的公司徽标。
.footer_right	页脚中的一个分割区域，例如，您可以使用此分割区域来显示其他徽标。

菜单栏

样式名	描述
.menubar	JSF 子视图。
.menuContainer	包含菜单项（例如标签和链接）的容器面板。
.menuItem	菜单栏中的项。

“登录”页面

样式名	描述
.loginPanel	包含登录表单的面板。
.loginTitle	表单的标题。
.loginText	指示型文本。
.loginForm	包含输入控件的表单。
.loginValues	用于确定控件布局的表。
.loginField	登录字段使用的标签，例如“姓名”或“密码”。
.loginValue	文本输入字段。

导航器

样式名	描述
.pageBodyNavigator	包含导航器的区域。
.navigator	包含列表链接的导航器的 JSF 子视图。
.navigatorTitle	每个导航器框的标题。
.taskNavigatorTitle	导航框的标题类。这些标题用于对业务流程对象列表链接与人员任务对象列表链接进行区分。
.navigatorFrame	每个导航器框的分割线，可用于绘制边框。
.navigatorLink	导航器框中的链接。
.expanded	当导航器框处于展开状态时使用。
.collapsed	当导航器框处于折叠状态时使用。

内容面板

样式名	描述
.pageBodyContent	包含内容的区域。

样式名	描述
.panelContainer	包含列表、详细信息或消息的分割面板。
.panelTitle	所显示的内容的标题，例如“我的待执行任务”。
.panelHelp	包含帮助文本和图标的分割容器。
.panelGroup	包含命令栏和列表以及详细信息或消息的分割容器。

命令栏

样式名	描述
.commandbar	命令栏区域外围的分割容器。
.button	用于显示命令栏按钮的样式。

列表

样式名	描述
.list	包含行的表。
.listHeader	列表中头行使用的样式。
.ascending	当列表根据此列的升序进行排序时的列表头类样式。
.descending	当列表根据此列的降序进行排序时的列表头类样式。
.unsorted	当列表未根据此列进行排序时的列表头类样式。

详细信息面板

样式名	描述
.details	详细信息面板外围的分割容器。
.detailsProperty	属性名的标签。
.detailsValue	属性值的文本。

消息数据

样式名	描述
.messageData	消息外围的分割容器。
.messageDataButton	消息表单中“添加”和“除去”按钮的按钮样式。
.messageDataOutput	用于显示只读文本。
.messageDataValidInput	用于有效的消息值。
.messageDataInvalidInput	用于无效的消息值。

选项卡式窗格

样式名	描述
.tabbedPane	所有选项卡式窗格外围的分割容器。
.tabHeader	选项卡式窗格的选项卡头。
.selectedTab	活动的选项卡头。

样式名	描述
.tab	不活动的选项卡头。
.tabPane	选项卡式窗格外围的分割容器。
.tabbedPaneNested	搜索页面上使用的嵌套型选项卡式窗格外围的分割容器。
.tabHeaderSimple	嵌套型选项卡式窗格的选项卡头。
tabHeaderProcess	流程过滤器的嵌套型选项卡式窗格的选项卡头。
.tabHeaderTask	任务过滤器的嵌套型选项卡式窗格的选项卡头。
.tabPaneSimple	嵌套型选项卡式窗格外围的分割容器。

搜索页面

样式名	描述
.searchPane	搜索面板的选项卡式窗格。另请参阅选项卡式窗格。
.searchPanelFilter	搜索表单的表容器。
.searchLabel	搜索表单控件的标签。
.summary	搜索总结窗格的外层容器。
.summaryTitle	搜索总结窗格上所有标题的公共样式。
.summaryTitleProcess	搜索总结窗格上与流程相关的部分的标题样式。
.summaryTitleTask	搜索总结窗格上与任务相关的部分的标题样式。

错误详细信息

样式名	描述
.errorPage	错误页面的选项卡式窗格。
.errorLink	用于在页面上显示按钮链接的样式。
.errorDetails	包含错误详细信息的选项卡式窗格。
.errorDetailsStack	包含异常堆栈信息的选项卡式窗格。
.errorDetailsMessage	错误消息的文本样式。

第 8 章 业务流程编排器观察器入门

在运行业务流程和任务时，WebSphere® Process Server 可以发出事件，这些事件包含流程实例及其相关活动的状态更改信息。您可以使用业务流程编排器观察器来根据这些事件检索统计信息并对流程和活动创建报告。

关于此任务

您可以定义自己的报告，也可以使用下寻方法来获取有关特定流程实例、活动实例或直接插入人员任务的更详细信息。此外，可以导出所报告的结果，以便进一步进行外部处理。

业务流程编排器观察器在 IT 级监视与业务级监视之间搭建了桥梁。它通过报告业务流程管理器组件中的事件，帮助您了解业务流程编排器中发生的情况。

相关概念

『业务流程编排器观察器用户界面』

业务流程编排器观察器是一个独立的 Web 应用程序，它提供了一组用于报告流程和活动事件的功能。业务流程编排器资源管理器用户界面由任务栏、导航窗格和工作空间组成。

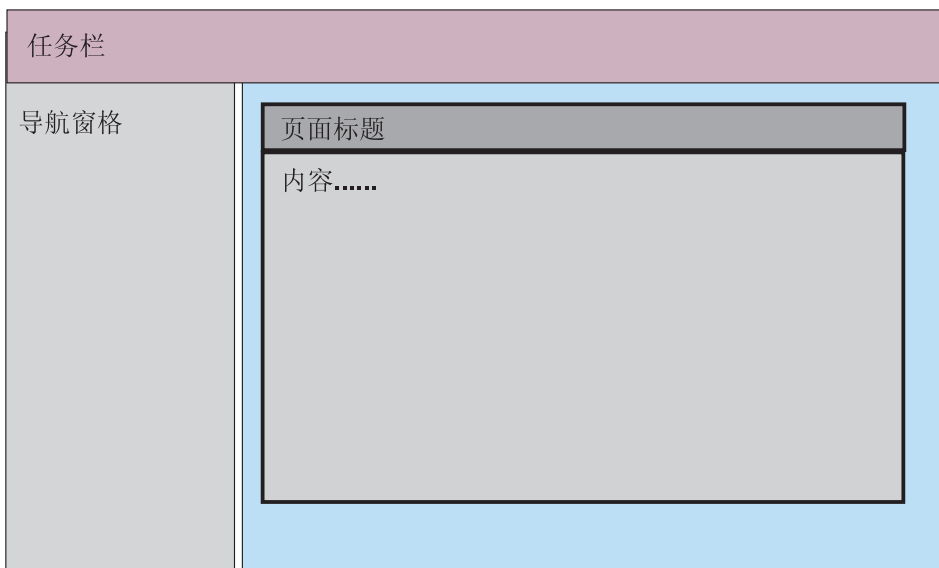
第 300 页的『业务流程编排器观察器的导航窗格』

使用导航窗格来选择所要创建的报告的类型，例如流程报告或活动报告。您还可以存储自己的报告定义并将它们添加到导航窗格。

业务流程编排器观察器用户界面

业务流程编排器观察器是一个独立的 Web 应用程序，它提供了一组用于报告流程和活动事件的功能。业务流程编排器资源管理器用户界面由任务栏、导航窗格和工作空间组成。

下图说明业务流程编排器观察器用户界面的布局。



用户界面的主要区域如下所示。


任务栏

任务栏提供用于从业务流程编排器观察器注销的选项以及指向常规帮助页面的链接。

导航窗格

用户界面左边的导航窗格包含链接，您可以使用这些链接来选择要创建的报告类型，例如，可以在图表中查看活动实例的数据。

工作空间

用户界面右边的工作空间包含一些页面，您可以使用这些页面来指定报告定义和查看报告。要访问这些页面，请单击导航窗格中的链接。要了解关于某个页面的信息，请单击相应页面上的帮助图标 。

相关任务

第 299 页的第 8 章，『业务流程编排器观察器入门』

在运行业务流程和任务时，WebSphere® Process Server 可以发出事件，这些事件包含流程实例及其相关活动的状态更改信息。您可以使用业务流程编排器观察器来根据这些事件检索统计信息并对流程和活动创建报告。

业务流程编排器观察器的导航窗格

使用导航窗格来选择所要创建的报告的类型，例如流程报告或活动报告。您还可以存储自己的报告定义并将它们添加到导航窗格。

可用的操作

在导航窗格中，可以执行下列操作：


- 折叠和展开组。

单击导航窗格中某个项旁边的加号 (+) 以展开该项，或者单击减号 (-) 以折叠该项。

- 转到预定义的列表或图表。

单击所要报告的实例的类型。

- 转到流程或活动报告向导。

单击**新建报告**图标 () 以指定报告的报告类型、报告内容和过滤器条件。

- 运行已保存的流程或活动报告。


单击报告名称，以运行该报告。

- 打开已保存的流程或活动报告定义的弹出菜单。


单击**显示弹出菜单**图标 () 以处理已保存的报告定义。


- 要编辑报告定义，请单击“编辑”图标 ()。

- 要复制报告定义，请单击“复制”图标 ()。

- 要删除报告定义，请单击“删除”图标 ()。

- 要导出报告结果，请单击“导出”图标 ()。

- 要以异步方式运行搜索，请单击**异步搜索**图标 ()。

- 在异步搜索成功完成后，在导航窗格中将显示**异步搜索完成**图标 ()。然后，请单击报告的名称以查看搜索结果。

- 如果异步搜索未成功完成，那么将显示**异步搜索失败**图标 ()。

导航窗格中的预定义列表和图表

导航窗格包含下列各组预定义列表和图表。

列表 这个组包含下列列表：

流程

使用此列表来查看在所指定时间范围内发出了流程事件的流程。这些流程将按流程状态列示。

活动

使用此列表来查看所选活动在所指定时间范围内进入的状态。这些活动将按活动状态列示。

用户

使用此列表来查看所选用户在所指定时间范围内执行的活动以及这些活动进入的状态。这些活动将按它们的状态显示。并且，还将显示每个活动的相应用户。

警告： 在某些国家或地区，收集有关职员工作表现的数据将违反隐私和数据保护法，在这种情况下，必须将流程模型定义成不发出各个用户的活动事件。

图表 这个组包含下列图表：

流程快照

使用此图表来检查在所指定时间处于不同状态的流程实例数。可以使用条形图或饼图来查看数据。

按时间段分组的流程

使用此图表来检查在所指定时间段内进入所指定状态的流程实例数的分布情况。每个实例都将显示在它进入所指定状态的时间片中。可以使用折线图、条形图或饼图来查看数据。


活动快照

使用此图表来检查在所指定时间处于不同状态的活动实例数。可以使用条形图或饼图来查看数据。

按时间段分组的活动

使用此图表来检查在所指定时间段内进入所指定状态的活动实例数的分布情况。每个实例都将显示在它进入所指定状态的时间片中。可以使用折线图、条形图或饼图来查看数据。

流程和活动报告

导航窗格链接至下列报告向导。报告向导由**新建报告**图标 () 指示。

流程报告

使用流程报告来查询流程实例事件。这些事件描述流程实例的状态更改。请使用报告向导来定义报告的数据。可以保存和检索报告定义。

活动报告

对于活动报告，可以查询活动实例事件。这些事件描述活动实例的状态更改。请使用报告向导来指定各个报告。可以存储和检索报告定义。

相关任务

第 299 页的第 8 章，『业务流程编排器观察器入门』

在运行业务流程和任务时，WebSphere® Process Server 可以发出事件，这些事件包含流程实例及其相关活动的状态更改信息。您可以使用业务流程编排器观察器来根据这些事件检索统计信息并对流程和活动创建报告。

第 9 章 管理业务流程和人员任务

业务流程和人员任务作为企业应用程序的一部分进行部署和安装。您可以使用管理控制台或管理命令来管理流程模板和任务模板，并可以使用业务流程编排器资源管理器来处理流程实例和任务实例。您可以使用业务流程编排器观察器来创建有关业务流程和人员任务的报告。

相关概念

第 3 页的第 1 章，『关于业务流程』

业务流程是一组与业务相关的活动，您通过调用这些活动来实现业务目标。

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

管理流程模板和流程实例

使用管理控制台或管理命令来管理流程模板。使用业务流程编排器资源管理器来处理流程实例。

关于此任务

流程模板定义企业应用程序中的业务流程。安装、部署和启动包含流程模板的企业应用程序时，流程模板将进入“已启动”状态。可以使用管理控制台或管理命令来停止和启动流程模板。已启动的流程模板将显示在业务流程编排器资源管理器中。

流程实例可以是长时间运行流程或微流程。使用业务流程编排器资源管理器来显示关于流程模板和流程实例的信息或者对流程实例执行操作。例如，这些操作可以是启动流程实例；对于长时间运行流程而言，可以是除流程生命周期操作以外的操作，例如暂挂、恢复或终止流程实例。此外，还可以执行修复活动。

相关概念

第 3 页的第 1 章，『关于业务流程』

业务流程是一组与业务相关的活动，您通过调用这些活动来实现业务目标。

『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

业务流程管理 - 常见问题

本节提供一组常见业务流程管理问题的回答。

- 第 304 页的『当流程模板处于“已启动”状态，但其所属应用程序处于“已停止”状态时，将出现什么情况？』
- 第 304 页的『如何停止正在创建的新流程实例？』
- 第 304 页的『当更新的流程模板生效时，对运行中的实例有何影响？』
- 第 304 页的『当模板停止时，对根据该模板创建的运行中实例有何影响？』
- 第 304 页的『如何确定是否有任何流程实例仍在运行？』
- 第 304 页的『为何无法停止包含流程实例的业务流程应用程序？』

当流程模板处于“已启动”状态，但其所属应用程序处于“已停止”状态时，将出现什么情况？

如果当前有效的流程模板处于“已启动”状态，但该应用程序处于“已停止”状态，那么不会根据该模板创建新的流程实例。当应用程序处于“已停止”状态时，无法对现有流程实例进行导航。

如何停止正在创建的新流程实例？

使用管理控制台，选择流程模板，然后单击**停止**。此操作将使该流程模板进入“已停止”状态，因此不会再根据该模板创建其他实例。在该模板停止后，任何根据该模板创建流程实例的尝试都将导致 `EngineProcessModelStoppedException` 错误。

当更新的流程模板生效时，对运行中的实例有何影响？

当某个流程模板不再有效时，并不会对运行中的根据该模板实例化的实例产生影响。现有流程实例将继续运行，直到完成为止。旧实例与新实例将并行地运行，直到所有旧实例完成或者被终止为止。

当模板停止时，对根据该模板创建的运行中实例有何影响？

将流程模板的状态更改为“已停止”时，仅停止正在创建的新实例。现有流程实例将继续有条理地运行，直到完成为止。

如何确定是否有任何流程实例仍在运行？

以流程管理员身份登录到业务流程编排器资源管理器，然后转到“我管理的流程实例”页面。此页面显示了任何运行中的流程实例。必要时，可以终止并删除这些流程实例。

为何无法停止包含流程实例的业务流程应用程序？

要运行流程实例，还必须运行相应的应用程序。如果将该应用程序停止，那么无法继续对该流程实例进行导航。因此，只能停止不包含流程实例的业务流程应用程序。

相关任务

第 303 页的『管理流程模板和流程实例』

使用管理控制台或管理命令来管理流程模板。使用业务流程编排器资源管理器来处理流程实例。

第 305 页的『使用管理控制台停止和启动流程模板』

可以使用管理控制台来单独启动和停止每个已安装的流程模板。

第 306 页的『使用管理命令停止和启动流程模板』

可以使用管理命令来代替使用管理控制台停止和启动流程模板。本节阐述如何使用管理命令来停止企业应用程序中的所有流程模板。

第 307 页的『管理流程的生命周期』

流程启动后，它将进入各种状态并最终结束。作为流程管理员，可以对处于生命周期各个阶段的流程执行各种操作。

第 307 页的『启动新的流程实例』

您可以根据有权使用的流程模板来启动新的流程实例。

第 308 页的『监视流程实例的进度』

可以对流程实例的进度进行监视，以确定是否需要执行操作以使该流程能够运行到完成。

第 308 页的『暂挂和恢复流程实例』

可以将顶层长时间运行流程实例暂挂。例如，通过执行此操作，可以配置对该流程稍后要使用的后端系统的访问或者解决导致流程实例失败的问题。满足该流程的先决条件后，可以恢复该流程实例。

第 309 页的『终止流程实例』

在某些情况下，您可能想终止流程实例，例如，不再需要该流程实例所代表的工作或文档、没有任何用户能够完成该流程实例以及流程模板有问题并需要进行重新设计等等。

第 310 页的『删除流程实例』

可以将流程模板设计成在流程实例完成时不自动将它们删除。在这些流程实例完成后，您可以手动将其删除。

第 311 页的『修复流程和活动』

如果流程遇到问题，您可以对其进行分析，然后修复活动。

第 311 页的『重新启动活动』

在修复活动之后，可以使用新的输入数据来将其重新启动。

第 312 页的『强制完成活动』

如果您知道活动无法及时完成，例如由于所调用的服务不再可用而无法及时完成，那么可以强制完成该活动以使流程可以继续运行。

第 312 页的『管理微流程的补偿』

微流程运行期间，它可能会遇到问题。对于这些情况，在流程模型中可能为流程定义了补偿。补偿允许您撤销先前已完成的步骤，例如，以便复位数据和状态，从而能够从这些问题中恢复。

使用管理控制台停止和启动流程模板

可以使用管理控制台来单独启动和停止每个已安装的流程模板。

开始之前

如果已启用 WebSphere 管理安全性，那么请验证您的用户标识是否具有操作员权限。安装了该应用程序的服务器必须处于运行状态。

关于此任务

例如，必须先停止流程模板，然后才能卸载该流程模板所属的业务流程应用程序。下列步骤描述如何使用管理控制台停止和启动流程模板。

过程

1. 选择要管理的模块。

在管理控制台的导航窗格中，单击**应用程序** → **SCA 模块** → **module_name**。

2. 在 EJB 模块的“配置”页面中，在**其他属性**中单击**业务流程**，然后单击流程模板。
3. 停止该流程模板。

该流程模板的现有实例将继续运行到正常结束为止。但是，不能根据已停止的模板来创建流程实例。

4. 启动处于“已停止”状态的流程模板。

相关概念

第 3 页的第 1 章, 『关于业务流程』

业务流程是一组与业务相关的活动, 您通过调用这些活动来实现业务目标。

第 303 页的 『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

使用管理命令停止和启动流程模板

可以使用管理命令来代替使用管理控制台停止和启动流程模板。本节阐述如何使用管理命令来停止企业应用程序中的所有流程模板。

开始之前

在开始执行本过程之前, 必须符合下列条件:

- 如果已启用 WebSphere 管理安全性, 那么请验证您的用户标识是否具有操作员权限。
- 要停止或启动的流程模板所在的应用程序服务器必须处于运行状态。即, 由于需要使用服务器连接, 因此不能使用 wsadmin 的 `-conntype none` 选项。

关于此任务

例如, 必须先停止流程模板, 然后才能卸载该流程模板所属的业务流程应用程序。下列步骤描述如何使用管理命令停止和启动流程模板。

过程

1. 切换到管理脚本所在的业务流程编排器子目录。

在 Windows 系统上, 输入:

```
cd install_root\ProcessChoreographer\admin
```

在 UNIX、Linux 和 i5/OS 系统上, 输入以下命令:

```
cd install_root/ProcessChoreographer/admin
```

2. 停止该流程模板。

在 Windows 系统上, 输入:

```
install_root\bin\wsadmin -f bpcTemplates.jacl  
                        -stop application_name
```

在 UNIX、Linux 和 i5/OS 系统上, 输入以下命令:

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -stop application_name
```

其中 `application_name` 是该模板所属应用程序的名称。

该流程模板的现有实例将继续运行到正常结束为止。应用程序停止后, 您将无法根据已停止的模板来创建流程实例。

3. 启动该流程模板。

在 Windows 系统上, 输入:

```
install_root\bin\wsadmin -f bpcTemplates.jacl  
                        -start application_name
```

在 UNIX、Linux 和 i5/OS 系统上，输入以下命令：

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -start application_name
```

该流程模板将启动。您可以使用业务流程编排器资源管理器来根据该流程模板启动流程实例。

相关概念

第 3 页的第 1 章，『关于业务流程』

业务流程是一组与业务相关的活动，您通过调用这些活动来实现业务目标。

第 303 页的『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

管理流程的生命周期

流程启动后，它将进入各种状态并最终结束。作为流程管理员，可以对处于生命周期各个阶段的流程执行各种操作。

相关概念

第 3 页的第 1 章，『关于业务流程』

业务流程是一组与业务相关的活动，您通过调用这些活动来实现业务目标。

第 303 页的『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

启动新的流程实例

您可以根据有权使用的流程模板来启动新的流程实例。

关于此任务

所有已安装的流程模板都将显示在业务流程编排器资源管理器的流程模板列表中。要启动新的流程实例，请完成下列步骤。

过程

1. 显示您有权使用的流程模板。

在导航窗格中的“流程模板”中，单击**我的流程模板**。

2. 选中该流程模板旁边的复选框，然后单击**启动实例**。

此操作将显示“流程输入消息”页面。

如果该流程有多个操作，那么此操作将显示包含所有可用操作的页面。请选择用于启动流程实例的操作。

3. 提供输入数据，以启动该流程实例。

如果该流程是长时间运行流程，那么可以输入流程实例名。如果未指定名称，那么将对新流程实例指定系统生成的名称。

填写流程输入消息的输入。

4. 要启动该流程，请单击**提交**。

结果

该流程实例已启动。如果该业务流程包含要求人员进行交互的活动，那么将生成可以由任何潜在所有者声明的任务。如果您是其中一位潜在所有者，那么此任务将会显示在“我的待执行任务”页面上的列表中。

如果该流程实例是微流程，那么该流程完成后将立即显示流程输出消息。对于长时间运行流程而言，请确保该流程实例不会在流程完成后被自动删除。在该情况下，流程输出消息将显示在“流程实例”视图中。并非所有流程都有输出消息。例如，如果该流程实现了单向操作，将不会显示输出消息。

相关概念

第 3 页的第 1 章，『关于业务流程』

业务流程是一组与业务相关的活动，您通过调用这些活动来实现业务目标。

第 303 页的『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

监视流程实例的进度

可以对流程实例的进度进行监视，以确定是否需要执行操作以使该流程能够运行到完成。

关于此任务

在业务流程编排器资源管理器中，请完成下列步骤以监视流程实例的进度。

过程

1. 显示流程实例列表。

例如，在导航窗格中，单击“流程实例”中的**由我管理**。

2. 选中该流程实例旁边的复选框，然后单击**查看流程状态**。

将显示“流程状态”页面。此页面显示了为流程定义的活动。这些活动是一些链接，其中包括链接的过渡和连接条件、故障处理程序、补偿处理程序以及事件处理程序。以粗体显示的活动在流程模型中被定义为与业务相关。对于这些活动，显示了状态信息。

3. 要对某个活动执行操作，请单击该活动。

将显示“活动”页面，您可以在该页面中执行操作，以使该流程能够运行到完成。

相关概念

第 3 页的第 1 章，『关于业务流程』

业务流程是一组与业务相关的活动，您通过调用这些活动来实现业务目标。

第 303 页的『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

暂挂和恢复流程实例

可以将顶层长时间运行流程实例暂挂。例如，通过执行此操作，可以配置对该流程稍后要使用的后端系统的访问或者解决导致流程实例失败的问题。满足该流程的先决条件后，可以恢复该流程实例。

开始之前

要暂挂和恢复流程实例，您必须具有流程管理员权限。

要暂挂某个流程实例，该流程实例必须处于“正在运行”或“即将失败”状态。要恢复某个流程，该流程实例必须处于“已暂挂”状态。

关于此任务

要暂挂或恢复某个流程实例，请在业务流程编排器资源管理器中完成下列步骤。

过程

1. 显示流程实例列表。

例如，在导航窗格中，单击“流程实例”中的**由我管理**。

2. 暂挂该流程。

选中该流程实例旁边的复选框，然后单击**暂挂**。

3. 选择其中一个选项以暂挂该流程实例。

- 要将该流程暂挂到您手动将其恢复为止，请选择**暂挂**。
- 要将该流程暂挂到特定时间为止，请选择**暂挂流程直至**，然后指定日期和时间。
- 要将该流程暂挂一段时间，请选择**暂挂流程**，然后指定持续时间。

4. 要确认您所作的选择，请单击**提交**。

此操作将暂挂指定的顶层流程实例。该流程实例将进入“已暂挂”状态。将 `autonomy` 属性设置为 `child` 的子流程如果处于“正在运行”、“即将失败”、“正在终止”或“正在补偿”状态，那么它们也将被暂挂。但是，仍可以完成任何属于该流程实例并处于运行状态的活动和任务。

下一步做什么？

要恢复处于“已暂挂”状态的流程实例，请选择该流程实例，然后单击**恢复**。该流程实例及其子流程将进入它们暂挂前所处的状态，例如“正在运行”。该流程实例及其子流程将继续运行。

相关概念

第 3 页的第 1 章，『关于业务流程』

业务流程是一组与业务相关的活动，您通过调用这些活动来实现业务目标。

第 303 页的『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

终止流程实例

在某些情况下，您可能想终止流程实例，例如，不再需要该流程实例所代表的工作或文档、没有任何用户能够完成该流程实例以及流程模板有问题并需要进行重新设计等等。

开始之前

要终止流程实例，您必须具有流程管理员权限。

关于此任务

在业务流程编排器资源管理器中，请完成下列步骤以终止流程实例。如果为业务流程模型定义了补偿，那么可以选择在进行补偿的情况下终止流程实例。

过程

1. 显示您可以管理的流程实例。

在导航窗格中的“流程实例”中，单击**由我管理**。

2. 选中要停止的流程实例旁边的复选框。

- 要在进行补偿的情况下终止该流程实例，请单击**补偿**。

此操作将终止该流程实例并启动补偿处理。

- 要在不进行补偿的情况下终止该流程实例，请单击**终止**。

此操作将立即停止该流程实例，而不会等待任何未完成的活动或任务完成。因此，不会对所终止的流程实例进行补偿。

相关概念

第 3 页的第 1 章，『关于业务流程』

业务流程是一组与业务相关的活动，您通过调用这些活动来实现业务目标。

第 303 页的『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

删除流程实例

可以将流程模板设计成在流程实例完成时不自动将它们删除。在这些流程实例完成后，您可以手动将其删除。

开始之前

要删除流程实例，您必须具有流程管理员权限。该流程实例必须处于“已完成”、“已失败”、“已终止”或“已补偿”状态。

关于此任务

如果在流程模型中设置了流程模板的相应属性，那么将从业务流程编排器数据库中自动删除已完成的流程实例。

您可能想将流程实例保留在数据库中，例如，以便查询未写入审计日志的流程实例数据或者将流程的删除工作推迟到非高峰期进行。但是，不再需要的旧流程实例数据会占用磁盘空间并影响性能。因此，应该定期删除不再需要的或者要进行维护的流程实例数据。请确保在非高峰期运行此维护任务。

可以使用业务流程编排器资源管理器来删除已完成的流程实例（例如，以删除个别流程实例），也可以使用 `deleteCompletedProcessInstances` 管理脚本同时删除多个流程实例。

在业务流程编排器资源管理器中，请完成下列步骤以删除流程实例。

过程

1. 显示您所管理的流程实例。

在导航窗格中的“流程实例”中，单击**由我管理**。

2. 选择要删除的流程实例，然后单击**删除**。

结果

此操作将从数据库中删除所选流程实例。

相关概念

第 3 页的第 1 章,『关于业务流程』

业务流程是一组与业务相关的活动,您通过调用这些活动来实现业务目标。

第 303 页的『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

修复流程和活动

如果流程遇到问题,您可以对其进行分析,然后修复活动。

关于此任务

业务流程编排器资源管理器为流程管理员提供了各种用于监视当前处于运行状态的流程的视图。

- 要查看活动处于“已停止”状态的流程实例,请在导航窗格的“流程实例”中单击**关键流程**。
- 要监视特定流程实例的进度,请在任何显示了流程实例列表的视图中单击**查看流程状态**。

下一步做什么?

现在,可以执行操作以修复暂挂的活动。

相关概念

第 3 页的第 1 章,『关于业务流程』

业务流程是一组与业务相关的活动,您通过调用这些活动来实现业务目标。

第 303 页的『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

重新启动活动

在修复活动之后,可以使用新的输入数据来将其重新启动。

开始之前

该活动必须处于“已停止”状态,相关联的流程实例必须处于“正在运行”状态。

关于此任务

要重新启动活动,请在业务流程编排器资源管理器中完成下列步骤。

过程

1. 转到该活动的“活动”页面,然后单击**重新启动**。

例如,在由我管理的流程实例页面上,单击流程实例的名称。在流程实例页面上,单击**活动**选项卡,然后单击要重新启动的活动的名称。

2. 指定用于再次启动活动的输入数据。

再次启动活动时,如果希望流程在出错时继续,请选中**出错时继续**。

3. 如果设置了活动的到期时间,那么请指定重新启动的活动的到期行为。

4. 单击重新启动。

相关概念

第 3 页的第 1 章,『关于业务流程』

业务流程是一组与业务相关的活动,您通过调用这些活动来实现业务目标。

第 303 页的『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

强制完成活动

如果您知道活动无法及时完成,例如由于所调用的服务不再可用而无法及时完成,那么可以强制完成该活动以使流程可以继续运行。

开始之前

通常,活动必须处于“已停止”状态。但是,如果该活动是人员活动,那么还可处于“就绪”或“已声明”状态。相关联的流程实例必须处于“正在运行”状态。

关于此任务

要强制完成活动,请在业务流程编排器资源管理器中完成下列步骤。

过程

1. 转到该活动的“活动”页面,然后单击**强制完成**。
2. 指定用于完成该活动的数据。
3. 再次单击**强制完成**。

相关概念

第 3 页的第 1 章,『关于业务流程』

业务流程是一组与业务相关的活动,您通过调用这些活动来实现业务目标。

第 303 页的『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

管理微流程的补偿

微流程运行期间,它可能会遇到问题。对于这些情况,在流程模型中可能为流程定义了补偿。补偿允许您撤销先前已完成的步骤,例如,以便复位数据和状态,从而能够从这些问题中恢复。

开始之前

要对微流程进行补偿,必须在管理控制台中启动补偿服务。

关于此任务

如果微流程的补偿操作失败,那么流程管理员必须进行干预以解决问题。

在业务流程编排器资源管理器中,请完成下列步骤以管理已失败的补偿操作。

过程

1. 显示已失败的补偿操作列表。

在导航窗格中的“流程实例”中,单击**失败的补偿**。

将显示“失败的补偿”页面。此页面包含所指定补偿操作的失败原因信息。此信息可以帮助您确定已失败的补偿的更正操作。

2. 选中该活动旁边的复选框，然后单击其中一个可用的操作。

提供了下列管理操作：

跳过 跳过当前补偿操作并继续对该微流程进行补偿。此操作可能会产生未补偿的活动。

重试 如果已执行了操作来更正已失败的补偿操作，那么可单击**重试**以再次尝试执行补偿操作。

停止 停止补偿处理。

相关概念

第 3 页的第 1 章，『关于业务流程』

业务流程是一组与业务相关的活动，您通过调用这些活动来实现业务目标。

第 303 页的『业务流程管理 - 常见问题』

本节提供一组常见业务流程管理问题的回答。

管理任务模板和任务实例

使用管理控制台或管理命令来管理任务模板。使用业务流程编排器资源管理器来处理任务实例。

相关概念

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

相关任务

第 285 页的第 7 章，『业务流程编排器资源管理器入门』

根据用户角色的不同，您可以使用业务流程编排器资源管理器来管理业务流程和人员任务，也可以处理分配给您的任务。

使用管理控制台停止和启动任务模板

使用管理控制台逐个启动和停止每个已安装的任务模板。

开始之前

如果已启用 WebSphere 管理安全性，那么请验证您的用户标识是否具有操作员权限。

关于此任务

任务模板用于定义服务组件体系结构（SCA）服务，这些服务作为企业应用程序中的独立任务表示。安装、部署和启动包含任务模板的企业应用程序时，任务模板将进入“已启动”状态。

过程

1. 选择要管理的模块。

在管理控制台的导航窗格中，单击**应用程序** → **SCA 模块** → **module_name**。

2. 在 EJB 模块的“配置”页面中，在**其他属性**中单击**人员任务**，然后单击流程模板。
3. 要停止该任务模板，请单击**停止**。

4. 要启动该任务模板，请单击 **启动**。

相关概念

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

使用管理命令停止和启动任务模板

可以使用管理命令来代替使用管理控制台停止和启动任务模板。使用管理命令来停止企业应用程序中的所有任务模板。

开始之前

如果已启用 WebSphere 管理安全性，那么请验证您是否使用了具有操作员权限的用户标识登录。

关于此任务

任务模板用于定义服务组件体系结构（SCA）服务，这些服务作为企业应用程序中的独立任务表示。安装、部署和启动包含任务模板的企业应用程序时，任务模板将进入“已启动”状态。

过程

1. 切换到管理脚本所在的业务流程编排器子目录。

在 Windows 系统上，输入：

```
cd install_root\ProcessChoreographer\admin
```

在 UNIX、Linux 和 i5/OS 系统上，输入以下命令：

```
cd install_root/ProcessChoreographer/admin
```

2. 停止该任务模板。

在 Windows 系统上，输入：

```
install_root\bin\wsadmin -f bpcTemplates.jacl  
                        -stop application_name
```

在 UNIX、Linux 和 i5/OS 系统上，输入以下命令：

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -stop application_name
```

其中 *application_name* 是该模板所属应用程序的名称。该任务模板的现有实例将继续运行到正常结束为止。

3. 启动该任务模板。

在 Windows 系统上，输入：

```
install_root\bin\wsadmin -f bpcTemplates.jacl  
                        -start application_name
```

在 UNIX、Linux 和 i5/OS 系统上，输入以下命令：

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -start application_name
```

该任务模板将启动。您可以使用业务流程编排器资源管理器来处理与该任务模板相关联的任务实例。

相关概念

第 33 页的第 2 章,『关于人员任务』

人员任务是一个组件,它允许人员与服务进行交互。

创建和启动任务实例

您可以根据有权使用的任务模板来创建和启动任务实例。

关于此任务

所有已安装的任务模板都将显示在业务流程编排器资源管理器的任务模板列表中。要根据任务模板来创建和启动任务实例,请完成下列步骤。

过程

1. 显示您有权使用的任务模板。

在导航窗格中的“任务模板”中,单击**我的任务模板**。

2. 选中该任务模板旁边的复选框,然后单击**启动实例**。

此操作将显示“任务输入消息”页面。

3. 提供输入数据,以启动该任务实例。
4. 要启动该任务实例,请单击**提交**。

结果

该任务实例已可供处理。

相关概念

第 33 页的第 2 章,『关于人员任务』

人员任务是一个组件,它允许人员与服务进行交互。

相关任务

第 285 页的第 7 章,『业务流程编排器资源管理器入门』

根据用户角色的不同,您可以使用业务流程编排器资源管理器来管理业务流程和人员任务,也可以处理分配给您的任务。

处理任务

要处理某个任务,必须先声明该任务,然后执行完成该任务所需的操作。

关于此任务

如果某个任务处于就绪状态,并且您是该任务的潜在所有者或管理员,那么可以声明该任务。如果声明任务,那么您将成为该任务的所有者并负责完成该任务。

如果您对某些任务具有阅读者或编辑者角色,那么那些任务也将显示在任务列表中。

要使用业务流程编排器资源管理器来声明并完成任务,请完成下列步骤。

过程

1. 显示已分配给您的任务。

单击**任务实例** → **我的待执行任务**。

此操作将显示“我的待执行任务”页面，此页面列示了已分配给您的任务。

2. 声明要处理的任務。

选中该任务旁边的复选框，然后单击**处理**。

此操作将显示“任务消息”页面。

3. 提供用于完成该任务的信息。

如果需要中断工作，例如需要同事提供更多信息才能完成该任务，请单击**保存**以保存所作的更改。

4. 单击**完成**以使用所提供的信息来完成该任务。

结果

您完成的任務將處於“已完成”狀態。如果在未完成該任務的情況下停止，那麼該任務將保持處於“已聲明”狀態。

相關概念

第 33 頁的第 2 章，『關於人員任務』

人員任務是一個組件，它允許人員與服務進行交互。

相關任務

第 285 頁的第 7 章，『業務流程編排器資源管理器入門』

根據用戶角色的不同，您可以使用業務流程編排器資源管理器來管理業務流程和人員任務，也可以處理分配給您的任務。

暫掛和恢復任務實例

可以暫掛任務實例。例如，您可能想暫掛任務實例，以便解決導致該任務實例失敗的問題。滿足該任務的先決條件後，可以恢復運行該任務實例。

開始之前

要暫掛和恢復任務實例，您必須具有任務管理員權限。

要暫掛某個任務實例，該任務實例必須處於“正在運行”或“即將失敗”狀態。要恢復某個任務，該任務實例必須處於“已暫掛”狀態。

只有使用 WebSphere Application Server 簡單日曆的人員任務才支持暫掛任務。

關於此任務

要暫掛某個任務實例，請在業務流程編排器資源管理器中完成下列步驟。

過程

1. 顯示您可以管理的任務實例。

在導航窗格中的“任務實例”中，单击**由我管理**。

2. 在任務實例頁面上，单击**暫掛**。

3. 选择其中一个选项以暂挂该任务实例。
 - 要将该任务暂挂到您手动将其恢复为止，请选择**暂挂**。
 - 要将该任务暂挂到特定时间为止，请选择**暂挂任务直至**，然后指定日期和时间。
 - 要将该任务暂挂一段时间，请选择**暂挂任务**，然后指定持续时间。
4. 要确认您所作的选择，请单击**提交**。该任务实例将进入“已暂挂”状态。

下一步做什么？

要恢复处于“已暂挂”状态的任务实例，请单击**恢复**。

相关概念

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

管理人员任务的优先级

可以使用人员任务的优先级对任务进行过滤以及对任务列表进行排序。

关于此任务

要更改任务实例的优先级，请在业务流程编排器资源管理器中完成下列步骤。

过程

1. 显示任务实例列表。

例如，在导航窗格中单击“任务实例”中的**我的待执行任务**。

2. 选中该任务实例旁边的复选框，然后单击**更改优先级**。
3. 输入值，然后单击**提交**。

该任务实例的优先级将设置为新值。

下一步做什么？

要按优先级对任务列表进行排序，请单击表头中的箭头。

相关概念

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

管理工作分配

启动任务后，可能需要管理该任务的工作分配，例如，以更好地在工作组成员之间分配工作量。

关于此任务

工作项是由于特定原因而分配给某位人员或一组人员的业务实体，例如任务或流程实例。分配原因允许人员在业务流程方案中充当各种角色，例如潜在所有者、编辑器或管理员。

由于不同的人员可以具有不同的角色，所以一个任务实例可以有多个相关联的工作项。例如，John、Sarah 和 Mike 全都是某个任务实例的潜在所有者，而 Anne 是管理

员，并且为全部这四个用户生成了工作项。John、Sarah 和 Mike 在任务列表中只能看到他们自己的工作项。由于 Anne 是管理员，所以她有自己的任务工作项，并且能够管理为 John、Sarah 和 Mike 生成的工作项。

有时，在启动任务后可能需要更改任务分配，例如将某个工作项从最初所有者移交给别人或者指定您不在场期间的离开设置。您可能还需要创建其他工作项或删除不再需要的工作项。

相关概念

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

相关任务

第 285 页的第 7 章，『业务流程编排器资源管理器入门』

根据用户角色的不同，您可以使用业务流程编排器资源管理器来管理业务流程和人员任务，也可以处理分配给您的任务。

移交您拥有的任务

如果您是任务所有者，那么可能需要将该任务移交给另一个用户。例如，如果需要他人提供信息才能完成该任务时，就必须这样做。

关于此任务

在业务流程编排器资源管理器中，请完成下列步骤以移交您拥有的任务。

过程

1. 显示您拥有的任务。

在导航窗格的“任务实例”组中，单击**我的待执行任务**。

2. 选中要移交的任务旁边的复选框，然后单击**移交**。
3. 移交该任务。

在**新所有者**字段中，指定新任务所有者的用户标识，然后单击**移交**。只能将该任务移交给该任务的另一个潜在所有者或任务管理员。

结果

移交的任务将显示在新任务所有者的任务列表中。

相关概念

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

相关任务

第 320 页的『指定离开设置』

如果您打算离开办公室一段时间，那么请指定替代您完成任务的人员。

第 321 页的『指定用户的离开设置』

如果用户无法处理他们的任务，例如休病假，那么可以指定替代该用户执行任务的人员。

移交以您作为任务启动者、发起者或管理员的工作项

开始执行任务后，可能需要更改工作分配。例如，如果任务所有者正在度假，但必须在此人员归来前完成该任务，那么需要将工作项移交给另一个用户。移交工作项的方式取决于您具有的角色以及任务状态。

开始之前

要移交工作项，您必须具有下列其中一个角色，并且，根据分配原因的不同，该任务必须处于下列其中一种状态。

角色	分配原因	任务状态	可以将工作项移交下列用户角色：
所有者	所有者	已声明	潜在所有者和管理员。
启动者	启动者	已到期、已终止、已完成、已失败或正在运行	潜在启动者和管理员。
发起者	发起者	任何任务状态	潜在实例创建者和管理员。
发起者	潜在启动者	不活动	任何用户角色。
管理员	启动者	已到期、已终止、已完成、已失败或正在运行	启动者。
管理员	潜在启动者	不活动	潜在启动者。
管理员	阅读者或管理员	除不活动状态以外的其他任何状态	阅读者和管理员。
管理员	潜在所有者或编辑器	“就绪”或“已声明”	潜在所有者或编辑器。

关于此任务

在业务流程编排器资源管理器中，请完成下列步骤以移交工作项。

过程

1. 显示您可以管理的任务实例。

在导航窗格中的“任务实例”中，单击**由我管理**。

2. 显示任务实例的工作项。

在“由我管理的任务实例”页面中，选中该任务实例旁边的复选框，然后单击**工作项**。

3. 移交工作项。

- a. 在**新所有者**字段中，指定新工作项所有者的用户标识。
- b. 选择一个或多个工作项，然后单击**移交**。

结果

移交的工作项将与新工作项所有者一起显示在工作项列表中。

相关概念

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

相关任务

第 321 页的『指定用户的离开设置』

如果用户无法处理他们的任务，例如休病假，那么可以指定替代该用户执行任务的人员。

指定离开设置

如果您打算离开办公室一段时间，那么请指定替代您完成任务的人员。

开始之前

要执行本任务，需要用于执行替代功能的虚拟成员管理器（VMM）人员目录提供程序。

关于此任务

根据所应用的替代策略的不同，可以有一个或多个替代者在您离开后接收分配给您的工作。每个任务模板的替代策略可以不同。

过程

1. 在任务栏中，单击**我的替代者**。
2. 在我的替代者页面中，指定离开设置，然后单击**保存**。
 - a. 要启用离开设置，请选中**我已离开**复选框。
 - b. 在**我的替代者**字段中，输入替代者的用户标识，然后单击**添加**。
 - c. 可选：根据需要，添加其他替代者。根据所应用的替代策略的不同，可以有一个或多个替代者在您离开后接收分配给您的工作。每个任务模板的替代策略可以不同。
 - d. 可选：要从列表中除去替代者，请选择该替代者的用户标识，然后单击**除去**。要选择多个替代者，请按住 **Ctrl** 键。
3. 请 **BPESystemAdministrator** 刷新人员查询结果。

结果

在**我已离开**复选框保持选中期间，您的替代者将接收到分配给您的工作。

下一步做什么？

必须单独移交选中**我已离开**复选框前分配给您的工作。

相关概念

第 74 页的『缺席者替代』

替代功能允许您为自己或者所管理的小组成员指定离开设置。替代策略定义如何处理分配给不在场用户的任务和升级。

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

相关任务

第 318 页的『移交您拥有的任务』

如果您是任务所有者，那么可能需要将该任务移交给另一个用户。例如，如果需要他人提供信息才能完成该任务时，就必须这样做。

第 173 页的『配置虚拟成员管理器人员目录提供程序』

通过本任务来配置虚拟成员管理器（VMM）人员目录提供程序，业务流程编排器将使用此提供程序来执行人员分配，从而确定能够启动流程或者声明活动/任务的人员。缺省人员目录提供程序已可供使用，仅当您引入了定制人员分配条件时才需要对其进行配置。

第 261 页的『使用管理控制台刷新人员查询结果』

人员查询的结果是静态的。您可以使用管理控制台来刷新人员查询。

第 279 页的『使用管理命令刷新人员查询结果』

人员查询的结果是静态的。您可以使用管理命令来刷新人员查询。

指定用户的离开设置

如果用户无法处理他们的任务，例如休病假，那么可以指定替代该用户执行任务的人员。

开始之前

您必须具有 `TaskSystemAdministrator` 权限才能执行本任务。要执行本任务，需要用于执行替代功能的虚拟成员管理器（VMM）人员目录提供程序。

过程

1. 在任务栏中，单击**定义替代者**。
2. 在定义替代者页面中，指定离开设置，然后单击**保存**。
 - a. 输入用户的用户标识，以便指定该用户的离开设置。
 - b. 要启用离开设置，请选中**用户离开**复选框。
 - c. 在**用户的替代者**字段中，输入要指定的替代者的用户标识，然后单击**添加**。
 - d. 可选： 根据需要，添加其他替代者。 根据所应用的替代策略的不同，可以有一个或多个替代者在该用户离开后接收分配给该人员的工作。 每个任务模板的替代策略可以不同。
 - e. 可选： 要从列表中除去替代者，请选择该替代者的用户标识，然后单击**除去**。 要选择多个替代者，请按住 **Ctrl** 键。
3. 刷新人员查询结果。

结果

在**用户离开**复选框保持选中期间，那些替代者将接收到分配给该用户的工作。

下一步做什么？

必须单独移交选中**用户离开**复选框前分配给不在场用户的工作。

相关概念

第 74 页的『缺席者替代』

替代功能允许您为自己或者所管理的小组成员指定离开设置。替代策略定义如何处理分配给不在场用户的任务和升级。

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

相关任务

第 319 页的『移交以您作为任务启动者、发起者或管理员的工作项』

开始执行任务后，可能需要更改工作分配。例如，如果任务所有者正在度假，但必须在此人员归来前完成该任务，那么需要将工作项移交给另一个用户。移交工作项的方式取决于您具有的角色以及任务状态。

第 318 页的『移交您拥有的任务』

如果您是任务所有者，那么可能需要将该任务移交给另一个用户。例如，如果需要他人提供信息才能完成该任务时，就必须这样做。

第 173 页的『配置虚拟成员管理器人员目录提供程序』

通过本任务来配置虚拟成员管理器（VMM）人员目录提供程序，业务流程编排器将使用此提供程序来执行人员分配，从而确定能够启动流程或者声明活动/任务的人员。缺省人员目录提供程序已可供使用，仅当您引入了定制人员分配条件时才需要对其进行配置。

第 261 页的『使用管理控制台刷新人员查询结果』

人员查询的结果是静态的。您可以使用管理控制台来刷新人员查询。

第 279 页的『使用管理命令刷新人员查询结果』

人员查询的结果是静态的。您可以使用管理命令来刷新人员查询。

创建工作项

您可能想为新的潜在所有者创建工作项。例如，在任何当前潜在所有者均无法接受更多的工作时，就可能需要执行此操作。此外，如果对人员目录执行的查询未返回任何潜在所有者，那么也可能需要创建工作项。例如，在长时间运行流程中，如果公司在流程启动后发生更改，那么可能会发生找不到潜在所有者的情况。

开始之前

要创建任务实例的工作项，您必须是该任务的适当角色。如果您是任务管理员，并且该任务处于下列其中一种状态，那么可以创建该任务实例的工作项：就绪、已声明、正在运行、已完成或已失败。如果该任务实例从任务模板派生，那么当该任务处于“已终止”或“已到期”状态时，您也可以创建工作项。

关于此任务

在业务流程编排器资源管理器中，请完成下列步骤以创建工作项。

过程

1. 显示您所管理的任务实例。

在导航窗格中的“任务实例”中，单击**由我管理**。

2. 选中任务实例旁边的复选框以便为该任务实例创建工作项，然后单击**创建工作项**。将显示创建工作项页面。
3. 创建工作项。
 - a. 在**新所有者**字段中，指定新工作项所有者的用户标识。
 - b. 从**原因**列表选择一个或多个角色。

这些角色确定所指定人员可以对新工作项执行的操作。

- c. 单击**创建**。

结果

将为您对新工作项所有者指定的每个角色创建工作项。新任务将显示在对此人员指定的任务列表中。

相关概念

第 33 页的第 2 章,『关于人员任务』

人员任务是一个组件,它允许人员与服务进行交互。

删除工作项

您可能想删除工作项。例如,如果错误地创建工作项,或者工作项的所属用户已离职,那么可以执行此操作。

开始之前

要删除任务实例的工作项,您必须是该任务的适当角色。如果您是任务管理员,那么可以在任务实例处于下列其中一种状态时删除工作项:就绪、已声明、正在运行、已完成或已失败。如果该任务实例从任务模板派生,那么您还可以在该任务实例处于“已终止”或“已到期”状态时删除该工作项。

关于此任务

在业务流程编排器资源管理器中,请完成下列步骤以删除工作项。

过程

1. 显示您所管理的任务实例。

在导航窗格中的“任务实例”中,单击**由我管理**。

2. 显示任务实例的工作项。

在由我管理的任务实例页面中,选择任务实例,然后单击**工作项**。

3. 删除工作项。

选择一个或多个工作项,然后单击**删除**。

结果

已删除工作项。

相关概念

第 33 页的第 2 章,『关于人员任务』

人员任务是一个组件,它允许人员与服务进行交互。

查看任务升级

升级将用户可能无法及时完成所分配任务这一情况通知升级接收者。

关于此任务

当任务逾期时,它可能会引起升级。升级将导致下列操作:

- 创建新的工作项,例如,以便管理员执行操作来帮助解决问题。
- 如果配置人员任务容器时指定了电子邮件设置,那么将向指定的人员发送一封电子邮件,以便将已升级任务的情况通知那些人员。
- 调用事件通知处理程序。

过程

要查看升级，请在“任务实例”中单击**我的升级内容**。

- 要查看关于升级的信息，请单击升级标识。
- 要查看关于已升级任务的信息，请单击任务名。

相关概念

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

发送升级电子邮件

当任务逾期时，它可能会引起升级。可以将系统设置为向指定的人员发送电子邮件，以便将关于该升级的信息通知那些人员。

开始之前

下列规则适用于升级电子邮件：

- 人员目录提供程序必须支持指定电子邮件地址，例如轻量级目录访问协议（LDAP）或虚拟成员管理器。
- 不支持 **Everybody**、**Nobody**、**Group** 和 **Users by user ID** 人员分配条件。例如，应该改为使用 **User records by user ID**。

过程

1. 在 WebSphere Integration Developer 中，请在人员任务编辑器中对该任务执行下列操作。

- a. 在属性区域的**详细信息**选项卡中的任务设置下方，编辑人员目录（**JNDI 名称**）字段的值。

将此字段的值设置为下列其中一项：

- `bpe/staff/samplevmmconfiguration`
 - `bpe/staff/samplevmmconfiguration`
 - 您选择的人员目录配置名称（JNDI 名称）。
- b. 在属性区域的**详细信息**选项卡中的升级设置下方，将**通知类型**字段值设置为电子邮件。
 - c. 指定要发送的升级电子邮件的正文。

要插入变量，以便在文本中包括特定于任务的信息，请单击**添加变量**并从列表中选择合适的变量。在编辑器中，此变量显示在一对“%”字符之间，但在运行时环境中，发送电子邮件时将对其进行求值并替换。

如果未指定任何文本，那么将使用缺省消息文本。

2. 在 WebSphere Process Server 中，执行下列操作。
 - a. 确保已设置简单电子邮件传输协议（SMTP）主机。如果已启用认证功能，那么请设置 SMTP 主机的用户标识和密码。

在管理控制台中，单击**资源** → **邮件** → **邮件会话** → **HTMailSession_nodeName_serverName** 以检查此设置，或者，如果业务流

程编排器被配置在集群中，那么请单击**资源** → **邮件** → **邮件会话** → **HTMailSession_clusterName**。SMTP 主机在单元级定义。

- b. 确保配置人员任务管理器时指定的发件人电子邮件地址（发件人电子邮件地址）有效。

在管理控制台中，单击**服务器** → **应用程序服务器** → **server_name** 以检查此设置，或者，如果业务流程编排器被配置在集群中，那么请单击**服务器** → **集群** → **cluster_name**。在**配置**选项卡上的“业务集成”部分中，单击**业务流程编排器** → **人员任务管理器**。

下一步做什么？

如果发送升级电子邮件时发生了问题，请检查 SystemOut.log 文件中的错误消息。

相关概念

第 33 页的第 2 章，『关于人员任务』

人员任务是一个组件，它允许人员与服务进行交互。

在业务流程编排器资源管理器中创建和编辑定制属性

创建新的定制属性，以指定流程实例、活动实例或任务实例的其他属性。

关于此任务

要创建实例的定制属性，请在业务流程编排器资源管理器中完成下列步骤。

过程

1. 显示流程实例、活动实例或任务实例的列表，然后单击实例名以打开详细信息页面。

例如，要打开任务实例列表，请在导航窗格中单击“任务实例”中的**我的待执行任务**。

2. 在“定制属性”选项卡上，单击**添加**。
3. 在**属性名**字段中输入该定制属性的名称，并在**属性值**字段中输入值。
4. 可选：要添加其他定制属性，请转至步骤 2。
5. 可选：要除去新的定制属性，请单击该定制属性旁边的**删除**图标。
6. 可选：要更改某个定制属性的属性名或属性值，请单击该定制属性，然后输入新值。
7. 单击**保存**。保存定制属性后，就无法更改属性名，并且无法删除该定制属性。

创建业务流程报告和活动报告

在处理业务流程和活动期间，当流程、活动或任务切换状态时，可以生成事件。这些事件将存储下来，您可以使用业务流程编排器观察器根据这些事件创建报告，例如，以便分析流程瓶颈或者对活动中调用的服务的可靠性进行评估。

关于此任务

您可以使用预定义的报告，也可以创建用户定义的流程报告和活动报告。

相关概念

『快照报告』

使用快照报告来确定活动或流程在特定日期和时间的状态。

第 328 页的『时间段报告』

使用时间段报告来确定特定活动事件或流程事件在一段时间内的发生频率。

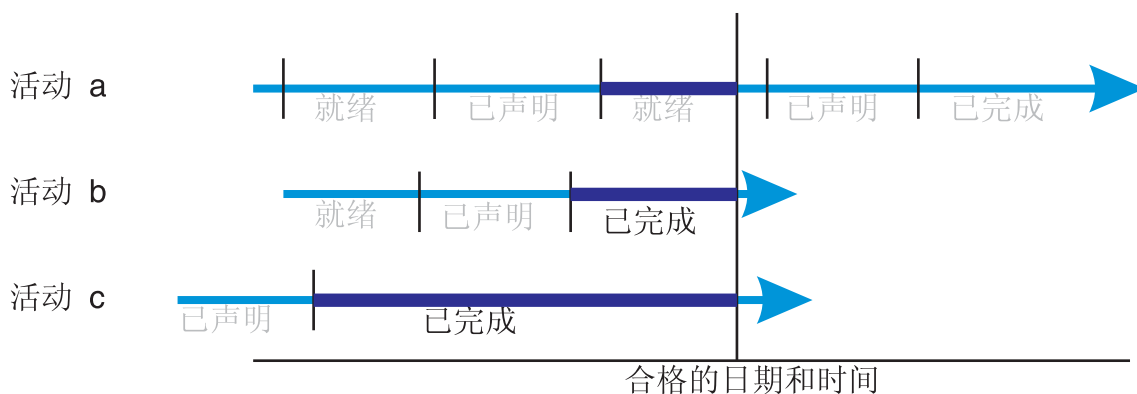
第 329 页的『时间处理』

在报告中，请考虑业务流程编排器观察器处理时间戳记和持续时间的方式。

快照报告

使用快照报告来确定活动或流程在特定日期和时间的状态。

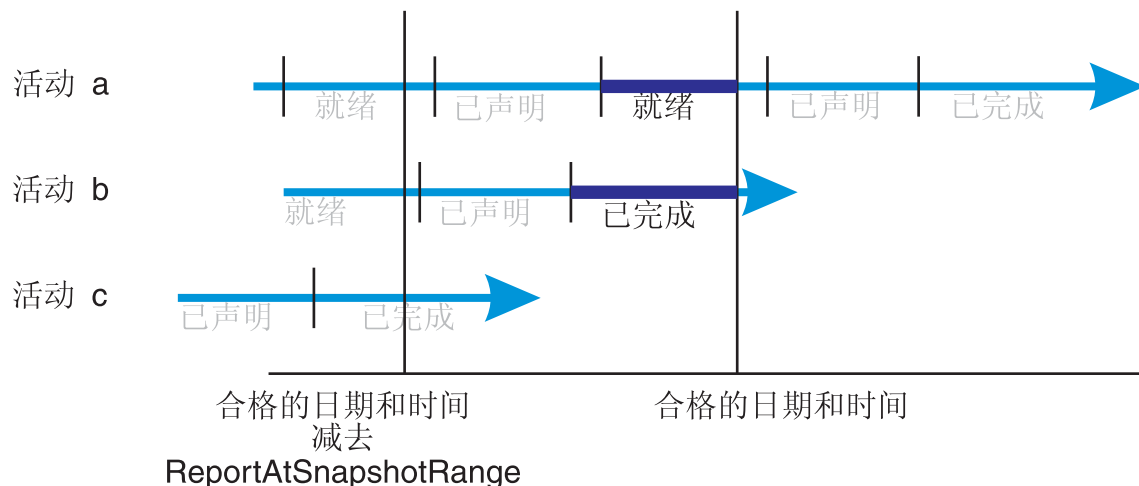
例如，您想要了解在午夜运行的流程实例数。对于每个流程实例或活动实例，业务流程编排器观察器将查找所指定日期和时间前的最后一个事件并评估结果状态。以下状态图显示符合快照报告条件的事件。



此快照包含一个处于“就绪”状态的活动（活动 a）和两个处于“已完成”状态的活动（活动 b 和 c）。

配置参数 ReportAtSnapshotRange

如果观察器数据库包含很长一段时间的流程实例数据，那么获取快照会比较耗时。为了避免查询不再相关的事件，请使用 `ReportAtSnapshotRange` 配置参数。报告将仅包含 `ReportAtSnapshotRange` 配置参数值与所指定日期和时间之间的数据。以下状态图显示设置 `ReportAtSnapshotRange` 参数后符合快照报告条件的事件。



此快照包含一个处于“就绪”状态的活动（活动 a）和一个处于“已完成”状态的活动（活动 b）。此报告未返回活动 c 的状态。

报告周期

您可以定义快照报告的报告周期。可以使用此选项来创建包含多个日期的重复快照的报告。例如，您想要报告三月份每一天启动的流程数。在这种情况下，不需要为每一天单独创建报告。而是，可以将开始日期定义为 3 月 1 日，将开始日期后的快照数定义为 31，并将快照之间的时间设置为 1 天。生成的报告将包含一个附加的列，以指示时间片编号。每个时间片的值都指示该月的日期。

相关任务

第 331 页的『创建预定义的快照图表』

使用预定义的快照图表来查看流程实例状态或活动实例状态在所指定日期和时间的分布情况。

第 334 页的『创建用户定义的快照报告』

您可以定义用户定义报告，以便获取所指定日期和时间的状态信息快照。您还可以创建包含报告时间段内定期时间点（例如，每个月第一天的午夜）的状态快照的报告。

第 234 页的『更改业务流程编排器观察器的配置参数』

对于启用验证功能以及提高性能而言，调整业务流程编排器观察器和事件收集器应用程序的配置参数至关重要。

第 325 页的『创建业务流程报告和活动报告』

在处理业务流程和活动期间，当流程、活动或任务切换状态时，可以生成事件。这些事件将存储下来，您可以使用业务流程编排器观察器根据这些事件创建报告，例如，以便分析流程瓶颈或者对活动中调用的服务的可靠性进行评估。

第 329 页的『使用预定义的列表和图表』

预定义的列表和图表使您能够采用下寻方法获取运行时实体的状态和事件信息。对于下寻过程中的每个步骤，您可以进一步定义所需的信息类型。例如，您可以指定日期和其他过滤器条件，以使用条形图来查看活动实例的数据。

第 334 页的『创建用户定义报告』

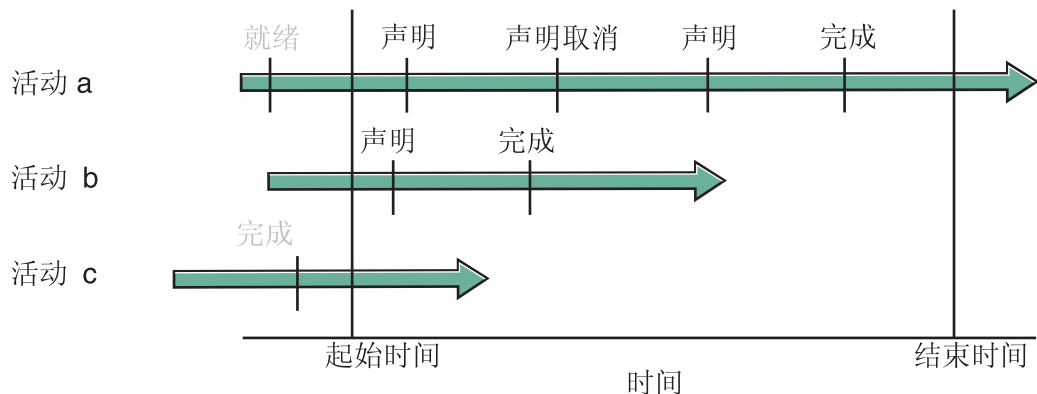
与预定义的列表和图表相比，用户定义的流程报告和活动报告更为灵活。此外，您可以存储和重复使用报告定义，并可以导出报告结果。

时间段报告

使用时间段报告来确定特定活动事件或流程事件在一段时间内的发生频率。

对于时间段视图，您可以指定报告时间段的开始日期和结束日期。该报告将包括这两个日期之间的时间间隔。例如，您想要了解当天声明的人员活动数。

以下状态图显示符合时间段报告条件的事件。以下示例所示时间段的报告包含 6 个活动事件，即活动 a 的 4 个事件以及活动 b 的 2 个事件。活动 c 在报告时间段开始前完成，因此该报告未包含该活动的事件。



这意味着，如果查询此时间段内完成的事件数，那么结果是 2。

报告周期

您可以定义时间段报告的报告周期。使用此选项来创建包含多个时间段的报告。例如，您想要报告过去 12 个月内每个月启动的流程数。在这种情况下，不需要为每个月单独创建报告。而是，可以将开始日期定义为 1 月 1 日，将开始日期后的时间片数定义为 12，并将时间片的长度定义为 1 个月。生成的报告将包含一个附加的列，以指示时间片编号。每个时间片的值都指示了月份。

相关任务

第 332 页的『创建预定义的时间段图表』

使用预定义的时间段图表来查看某个时间段内进入所指定状态的流程实例或活动实例的数目分布情况。每个实例都将显示在它进入所指定状态的时间片中。

第 337 页的『创建用户定义的时间段报告』

您可以为一段时间内发生的流程事件或活动事件创建用户定义报告。还可以创建根据报告周期涵盖多个时间段的报告。

第 325 页的『创建业务流程报告和活动报告』

在处理业务流程和活动期间，当流程、活动或任务切换状态时，可以生成事件。这些事件将存储下来，您可以使用业务流程编排器观察器根据这些事件创建报告，例如，以便分析流程瓶颈或者对活动中调用的服务的可靠性进行评估。

第 329 页的『使用预定义的列表和图表』

预定义的列表和图表使您能够采用下寻方法获取运行时实体的状态和事件信息。对于下寻过程中的每个步骤，您可以进一步定义所需的信息类型。例如，您可以指定日期和其他过滤器条件，以使用条形图来查看活动实例的数据。

第 334 页的『创建用户定义报告』

与预定义的列表和图表相比，用户定义的流程报告和活动报告更为灵活。此外，您可以存储和重复使用报告定义，并可以导出报告结果。

时间处理

在报告中，请考虑业务流程编排器观察器处理时间戳记和持续时间的方式。

时间戳记

在数据库中，时间戳记以全球标准时间（UTC）格式存储。输入和显示的时间戳记始终是用户界面运行位置的本地时间。这意味着，如果指定具有报告周期的快照报告，并且夏令时的时钟调整时刻在该报告周期内，那么日期和时间在时钟变更后将调整一小时。

例如，如果指定具有报告周期的快照报告，并且在冬季时间的上午 8:00 获取第一个快照，后续快照每 24 小时获取一次，那么在夏令时段，获取快照的时间为上午 9:00。

以月份和年份计的持续时间

如果指定具有报告周期的报告，并且时间片长度以月份或年份为单位，那么各个时间片的长度将随日历变化而变化。这允许您指定每个时间片都代表一个月的报告。

相关任务

第 325 页的『创建业务流程报告和活动报告』

在处理业务流程和活动期间，当流程、活动或任务切换状态时，可以生成事件。这些事件将存储下来，您可以使用业务流程编排器观察器根据这些事件创建报告，例如，以便分析流程瓶颈或者对活动中调用的服务的可靠性进行评估。

『使用预定义的列表和图表』

预定义的列表和图表使您能够采用下寻方法获取运行时实体的状态和事件信息。对于下寻过程中的每个步骤，您可以进一步定义所需的信息类型。例如，您可以指定日期和其他过滤器条件，以使用条形图来查看活动实例的数据。

第 334 页的『创建用户定义报告』

与预定义的列表和图表相比，用户定义的流程报告和活动报告更为灵活。此外，您可以存储和重复使用报告定义，并可以导出报告结果。

使用预定义的列表和图表

预定义的列表和图表使您能够采用下寻方法获取运行时实体的状态和事件信息。对于下寻过程中的每个步骤，您可以进一步定义所需的信息类型。例如，您可以指定日期和其他过滤器条件，以使用条形图来查看活动实例的数据。

关于此任务

提供了下列类型的预定义列表和图表：

- 列表
- 流程快照图表和活动快照图表
- 流程实例和活动实例时间段图表

相关概念

第 326 页的『快照报告』

使用快照报告来确定活动或流程在特定日期和时间的状态。

第 328 页的『时间段报告』

使用时间段报告来确定特定活动事件或流程事件在一段时间内的发生频率。

第 329 页的『时间处理』

在报告中，请考虑业务流程编排器观察器处理时间戳记和持续时间的方式。

相关任务

第 333 页的『示例：使用预定义的图表』

本场景提供一个示例，阐述如何使用预定义的图表。

『示例：使用预定义的列表』

本场景提供一个示例，阐述如何使用预定义的列表。

使用预定义的列表来创建报告

使用预定义的列表来报告指定时间段内发生的流程事件数或活动事件数，并按状态排序。还可以使用这些列表来下寻到特定实例的事件。此外，还可以导出每种状态的报告结果。

过程

1. 选择列表类型。

预定义的列表可用于与用户相关联的流程实例、活动实例和活动。

2. 输入您感兴趣的时间段的开始日期和结束日期，然后单击**继续**。

根据列表类型的不同，将显示流程模板列表、活动模板列表或者用户及其相关实例数的列表。

3. 选择您感兴趣的实例的复选框，然后单击**实例快照**。

在选项卡式窗格中将显示所选实例的事件。每个页面都显示了处于特定状态的实例。

4. 可选：要查看特定实例的所有事件以及获取更多有关该实例的信息，请单击实例名。
5. 可选：要以 CSV 格式导出所报告的数据，请单击**导出**。选择是要打开还是要保存所生成的导出数据，然后单击**确定**。系统将导出当前所显示状态的报告数据。

相关概念

第 329 页的『时间处理』

在报告中，请考虑业务流程编排器观察器处理时间戳记和持续时间的方式。

相关任务

『示例：使用预定义的列表』

本场景提供一个示例，阐述如何使用预定义的列表。

示例：使用预定义的列表

本场景提供一个示例，阐述如何使用预定义的列表。

开始之前

您的工厂生产三种不同的产品 Item1、Item2 和 Item3。制造和交货流程作为 WebSphere Process Server 中的 SOA 流程建模和运行。每个客户订单都由相应流程模板的专用流

程实例表示。将产品交付客户后，交付流程将进入最终状态“已完成”。如果客户取消订单，那么相应的流程实例将终止并进入“已终止”状态。

要查看上个月取消订购 Item1、Item2 或 Item3 的客户数，意味着您对进入“已终止”状态的流程实例数感兴趣。此外，您还想了解取消订单时该订单的处理进度。

关于此任务

使用预定义的列表来创建视图，以便显示已取消的流程数以及查看执行取消时该流程所处的状态：

过程

1. 在导航栏的列表部分中，选择**流程**。
2. 在搜索条件页面上，输入您感兴趣的时间段的开始日期和结束日期，然后单击**继续**。流程模板页面将列示所有在观察时间段内生成了流程的流程模板。对于每个流程模板，您可以查看已启动和结束的流程实例数。
3. 在流程模板页面上，选择该列表的所有模板，然后单击**实例快照**。流程实例页面将列示所有流程实例，这些流程实例将按它们在观察时间段内进入的状态分组。
4. 在流程实例页面上，选择**已终止**选项卡，以查看观察时间段内执行的取消操作的总数。
5. 按模板名称对此列表进行排序，确定每个流程模板的取消操作数。
6. 要获取更多详细信息，请单击已终止的流程实例的名称，以查看流程实例详细信息页面。然后，检查该实例的工作时间和耗用时间。

相关任务

第 329 页的『使用预定义的列表和图表』

预定义的列表和图表使您能够采用下寻方法获取运行时实体的状态和事件信息。对于下寻过程中的每个步骤，您可以进一步定义所需的信息类型。例如，您可以指定日期和其他过滤器条件，以使用条形图来查看活动实例的数据。

第 330 页的『使用预定义的列表来创建报告』

使用预定义的列表来报告指定时间段内发生的流程事件数或活动事件数，并按状态排序。还可以使用这些列表来下寻到特定实例的事件。此外，还可以导出每种状态的报告结果。

创建预定义的快照图表

使用预定义的快照图表来查看流程实例状态或活动实例状态在所指定日期和时间的分布情况。

开始之前

要查看图表，浏览器必须能够运行 Macromedia Flash Player。

过程

1. 选择快照图表的类型。

预定义的快照图表可用于流程实例和活动实例。

2. 输入搜索条件，然后单击**继续**。

将显示符合该搜索条件的对象模板列表。

3. 选择您感兴趣的模板的复选框，然后单击**继续处理所选模板**。

您可以更改图表类型，以便将结果显示为条形图或饼图。

相关概念

第 326 页的『快照报告』

使用快照报告来确定活动或流程在特定日期和时间的状态。

第 329 页的『时间处理』

在报告中，请考虑业务流程编排器观察器处理时间戳记和持续时间的方式。

相关任务

第 333 页的『示例：使用预定义的图表』

本场景提供一个示例，阐述如何使用预定义的图表。

创建预定义的时间段图表

使用预定义的时间段图表来查看某个时间段内进入所指定状态的流程实例或活动实例的数目分布情况。每个实例都将显示在它进入所指定状态的时间片中。

开始之前

要查看图表，浏览器必须能够运行 Macromedia Flash Player。

关于此任务

例如，您可以使用预定义图表来查看过去 12 个月内已完成流程实例的分布情况。

过程

1. 选择时间段图表的类型。

预定义的时间段图表可用于流程实例和活动实例。

2. 输入搜索条件，然后单击**继续**。

输入时间段的开始日期，然后指定时间片的数目、每个时间片的长度以及要报告的状态。例如，要报告过去 12 个月内每个月的已完成实例，请指定 12 作为时间片数目，并指定 1 个月作为每个时间片的长度。

将显示符合该搜索条件的对象模板列表。

3. 选择您感兴趣的模板的复选框，然后单击**继续处理所选模板**。

您可以更改图表类型，以便将结果显示为条形图、折线图或饼图。

相关概念

第 328 页的『时间段报告』

使用时间段报告来确定特定活动事件或流程事件在一段时间内的发生频率。

第 329 页的『时间处理』

在报告中，请考虑业务流程编排器观察器处理时间戳记和持续时间的方式。

相关任务

第 333 页的『示例：使用预定义的图表』

本场景提供一个示例，阐述如何使用预定义的图表。

示例: 使用预定义的图表

本场景提供一个示例, 阐述如何使用预定义的图表。

关于此任务

您的工厂生产两种不同的产品 Item1 和 Item2。制造和交货流程作为 WebSphere Process Server 中的 SOA 流程建模和运行。每个客户订单都由相应流程模板的专用流程实例表示。

最近, 您扩充了生产线, 推出了新产品 Item3。您有了新的 Item3 订购模板, 并且要了解该生产线上个月的进度。作为一项指标, 您想要查看最近 30 天内的生产订单数。

为了将最近 30 天内处理的生产订单数可视化, 请指定一个图表视图, 以显示所感兴趣的时间段内所有与 OrderItem3 流程模板相关的流程实例:

过程

1. 在导航栏的**图表**部分中, 选择“流程 (按时间段排列)”以查看最近 30 天内流程实例的统计分布。
2. 指定搜索条件:
 - a. 输入观察时间段的开始日期
 - b. 将时间片的数目设置为 30。
 - c. 将时间片的长度设置为 1 天。
 - d. 在**关注状态**列表中, 选择**正在运行**, 然后单击**继续**。

将打开选择流程模板页面, 此页面包含所有与观察时间段内发生的流程实例相关的流程模板。

3. 选择 OrderItem3 模板以查看所有与此流程模板相关的流程实例, 然后单击**继续处理所选模板**。
4. 流程实例快照页面显示了所有在指定时间处于不同状态的流程实例。
5. 使用折线图或条形图将该流程在上个月的进度可视化。

下一步做什么?

该报告将显示所有在观察时间段内进入“正在运行”状态的流程实例。

相关任务

第 329 页的『使用预定义的列表和图表』

预定义的列表和图表使您能够采用下寻方法获取运行时实体的状态和事件信息。对于下寻过程中的每个步骤, 您可以进一步定义所需的信息类型。例如, 您可以指定日期和其他过滤器条件, 以使用条形图来查看活动实例的数据。

第 331 页的『创建预定义的快照图表』

使用预定义的快照图表来查看流程实例状态或活动实例状态在所指定日期和时间的分布情况。

第 332 页的『创建预定义的时间段图表』

使用预定义的时间段图表来查看某个时间段内进入所指定状态的流程实例或活动实例的数目分布情况。每个实例都将显示在它进入所指定状态的时间片中。

创建用户定义报告

与预定义的列表和图表相比，用户定义的流程报告和活动报告更为灵活。此外，您可以存储和重复使用报告定义，并可以导出报告结果。

关于此任务

对于流程报告，您可以获取有关流程实例属性以及流程实例所包含活动的信息。对于活动报告，可以获取有关活动属性以及那些活动的相关流程实例的信息。您可以定义一次性的报告，也可以保存报告定义以便可以在需要时运行该报告。通过包括参数，可以在每次运行该报告时更改报告定义值。

相关概念

第 326 页的『快照报告』

使用快照报告来确定活动或流程在特定日期和时间的状态。

第 328 页的『时间段报告』

使用时间段报告来确定特定活动事件或流程事件在一段时间内的发生频率。

第 329 页的『时间处理』

在报告中，请考虑业务流程编排器观察器处理时间戳记和持续时间的方式。

相关任务

第 339 页的『示例：使用用户定义报告』

本场景提供一个示例，阐述如何使用用户定义报告。

相关参考

第 340 页的『报告属性』

使用属性来定义报告内容以及对结果进行过滤。可用的属性视报告类型而定。

第 342 页的『业务流程编排器观察器的业务流程事件』

如果您要求对 WebSphere Integration Developer 中的业务流程元素进行监视，那么将发送业务流程事件。在这些事件中，有一小部分事件可供业务流程编排器观察器使用。

第 343 页的『与性能相关的属性』

运行报告定义所需的时间会有所变化。为了提高生成报告时的性能，可以对报告定义进行优化。有一些通用规则可以帮助您评估报告属性对性能的影响。


创建用户定义的快照报告

您可以定义用户定义报告，以便获取所指定日期和时间的状态信息快照。您还可以创建包含报告时间段内定期时间点（例如，每个月第一天的午夜）的状态快照的报告。

关于此任务

报告向导将指导您定义报告。

过程

1. 在导航窗格中，对流程报告或活动报告单击**新建报告**图标 ()。
2. 在选择报告类型页面上，单击**快照报告**，然后单击**下一步**。
3. 在选择快照类型页面上，指定快照获取时间，然后单击**下一步**。
 - 要查看当前状态，请单击**立即获取快照**。每次运行报告时，都将对快照日期和时间进行求值。

将显示指定内容页面。转到步骤 5。

- 要查看流程或活动在特定日期和时间（例如 6 月 10 日上午 8 点）的状态，请单击**在特定日期和时间获取快照**。

将显示指定快照设置页面。转到步骤 4。

- 要查看报告时间段中定期时间点的状态，请单击**根据报告周期获取重复快照**。

将显示“指定快照设置”页面。转到步骤 4。

4. 指定快照设置，然后单击下一步。


如果要在特定日期和时间获取快照，请指定日期和时间设置。可以指定将来的日期和时间。要在每次运行报告时更改设置，请选中**使用这些设置作为参数复选框**。

对于具有报告周期的报告：

- a. 选择是要设置报告周期的开始日期还是结束日期，然后单击**下一步**。
 - b. 要设置报告周期的开始日期，请指定第一个快照的获取时间。要设置报告周期的结束日期，请指定最后一个快照的获取时间。
 - c. 要定义报告周期的持续时间，请设置快照数以及各个快照之间的间隔时间。
 - d. 要在每次运行报告时更改报告周期设置，请选中**使用这些设置作为参数复选框**。
5. 在指定报告内容页面上，指定要让该报告包含的信息，然后单击**下一步**。

对于具有报告周期的报告，属性列表已包含快照编号属性。您无法删除此属性。

- a. 单击**添加**以查看一组可以包括在报告中的属性；这些属性将成为该报告的列标题。属性的位置确定了该报告中各个列的顺序。对于每个属性，您还可以指定该列中结果的排序方式。如果对多个属性指定排序顺序，那么结果将按那些属性的顺序进行排序。请考虑重新安排属性的顺序，以更改该报告中结果的排序顺序。

- 要修改属性，请单击**修改**图标 ()。

- 要删除属性，请单击**删除**图标 ()。

- 要更改属性在报告中的位置，请单击**上移**图标 () 或**下移**图标 ()。

- b. 要对结果中的条目数进行限制（例如，以便提高性能），请在**阈值**字段中输入一个值，以指定最大结果数。


缺省阈值是 20。如果您不想对结果进行限制，请将此值设置为 -1。

要在每次运行报告时更改设置，请选中**使用阈值作为参数复选框**。

6. 可选：在指定过滤器内容页面上，设置属性的过滤器条件。

使用过滤器条件来限制属性可以具有的值，从而使报告更为具体。报告将仅包含那些满足所有指定过滤器条件的流程和活动。如果在指定报告内容页面上指定了聚集属性，那么过滤器条件列表已包含此属性的过滤器条件。您无法删除此过滤器。

- a. 单击**添加**以查看一组可以对其指定过滤器条件的属性。

- 对于更复杂的值类型，例如时间戳记，请单击**输入辅助控件**图标 () 来填写字段。
- 要在每次运行报告时更改过滤器条件值，请选中**参数**复选框。

b. 单击**下一步**。

将显示总结页面。此页面显示了报告定义。

7. 在总结页面上，执行下列其中一项操作：

- 如果报告定义未包含参数，那么请单击**运行**。

将显示所生成的报告。

- 如果报告定义包含参数，那么请单击**下一步**。

您可以更改参数值，然后单击**运行**。将显示所生成的报告。

如果报告结果与您的预期不符，那么可以单击**编辑**以更改报告的设置。

8. 可选： 导出报告结果。

要以 CSV 格式导出所报告的数据，请单击**导出**。选择是要打开所生成的导出数据还是要将其保存在硬盘上，然后单击**确定**。

仅当报告列表包含项时，才会显示**导出**按钮。

9. 可选： 保存报告定义。

如果要多次运行此报告，例如每个月运行一次此报告以显示每个月 10 号的已完成流程实例，那么请单击**保存**并输入报告名称。 该报告将显示在导航窗格中。

相关概念

第 326 页的『快照报告』

使用快照报告来确定活动或流程在特定日期和时间的状态。

第 329 页的『时间处理』

在报告中，请考虑业务流程编排器观察器处理时间戳记和持续时间的方式。

相关任务

第 339 页的『示例：使用用户定义报告』

本场景提供一个示例，阐述如何使用用户定义报告。

相关参考

第 340 页的『报告属性』

使用属性来定义报告内容以及对结果进行过滤。可用的属性视报告类型而定。

第 342 页的『业务流程编排器观察器的业务流程事件』

如果您要求对 WebSphere Integration Developer 中的业务流程元素进行监视，那么将发送业务流程事件。在这些事件中，有一小部分事件可供业务流程编排器观察器使用。

第 343 页的『与性能相关的属性』

运行报告定义所需的时间会有所变化。为了提高生成报告时的性能，可以对报告定义进行优化。有一些通用规则可以帮助您评估报告属性对性能的影响。


创建用户定义的时间段报告

您可以为一段时间内发生的流程事件或活动事件创建用户定义报告。还可以创建根据报告周期涵盖多个时间段的报告。

关于此任务

报告向导将指导您定义报告。

过程

1. 在导航窗格中，对流程报告或活动报告单击**新建报告**图标 ()。
2. 在选择报告类型页面上，单击**时间段报告**，然后单击**下一步**。
3. 在选择时间段类型页面上，指定时间段类型，然后单击**下一步**。

例如，对于流程，可以选择下列其中一种时间段类型：

- 要查看从指定日期到现在的事件，请单击**报告到现在为止的所有流程**。
- 要查看所指定时间段内的事件，请单击**报告指定时间段内的流程**。
- 要查看报告时间段内定期时间间隔的事件，请单击**按照报告周期报告流程**。

将显示指定日期和时间页面。

4. 指定日期和时间设置，然后单击**下一步**。





要报告到现在为止的所有流程，请指定开始日期。每次运行该报告时，都将生成结束日期。要报告指定时间段内的流程，请指定开始日期和结束日期。可以指定将来的日期。要在每次运行报告时更改设置，请选中**使用这些设置作为参数**复选框。

对于具有报告周期的报告：

- a. 选择是要设置报告周期的开始日期还是结束日期，然后单击**下一步**。
 - b. 要设置报告周期的开始日期，请指定第一个时间片的开始日期。要设置报告周期的结束日期，请指定最后一个时间片的结束日期。
 - c. 要定义报告周期的持续时间，请设置时间片的总数以及每个时间片的长度。
 - d. 要在每次运行报告时更改报告周期设置，请选中**使用这些设置作为参数**复选框。
5. 在指定报告内容页面上，指定要让该报告包含的信息，然后单击**下一步**。

对于具有报告周期的报告，属性列表已包含时间片编号属性。您无法删除此属性。

- a. 单击**添加**以查看一组可以包括在报告中的属性；这些属性将成为该报告的列标题。属性的位置确定了该报告中各个列的顺序。对于每个属性，您还可以指定该列中结果的排序方式。如果对多个属性指定排序顺序，那么结果将按那些属性的顺序进行排序。请考虑重新安排属性的顺序，以更改该报告中结果的排序顺序。

- 要修改属性，请单击**修改**图标 ()。
- 要删除属性，请单击**删除**图标 ()。
- 要更改属性在报告中的位置，请单击**上移**图标 () 或**下移**图标 ()。

- b. 要对结果中的条目数进行限制（例如，以便提高性能），请在**阈值**字段中输入一个值，以指定最大结果数。


缺省阈值是 20。如果您不想对结果进行限制，请将此值设置为 -1。

要在每次运行报告时更改设置，请选中**使用阈值作为参数**复选框。

6. 可选：在指定过滤器内容页面上，设置属性的过滤器条件。

使用过滤器条件来限制属性可以具有的值，从而使报告更为具体。如果在指定报告内容页面上指定了聚集属性，那么过滤器条件列表已包含此属性的过滤器条件。您无法删除此过滤器。

- a. 单击**添加**以查看一组可以对其指定过滤器条件的属性。

- 对于更复杂的值类型，例如时间戳记，请单击输入**辅助控件**图标（）来填写字段。
- 要在每次运行报告时更改过滤器条件值，请选中**参数**复选框。

- b. 单击**下一步**。

将显示总结页面。此页面显示了报告定义。

7. 在总结页面上，执行下列其中一项操作：

- 如果报告定义未包含参数，那么请单击**运行**。

将显示所生成的报告。

- 如果报告定义包含参数，那么请单击**下一步**。

您可以更改参数值，然后单击**运行**。将显示所生成的报告。

如果报告结果与您的预期不符，那么可以单击**编辑**以更改报告的设置。

8. 可选：导出报告结果。

要以 CSV 格式导出所报告的数据，请单击**导出**。选择是要打开所生成的导出数据还是要将其保存在硬盘上，然后单击**确定**。

仅当报告列表包含项时，才会显示**导出**按钮。

9. 可选：保存报告定义。

如果要定期运行此报告，例如每个月生成一次报告，那么请单击**保存**并输入报告名称。该报告将显示在导航窗格中。

相关概念

第 328 页的『[时间段报告](#)』

使用时间段报告来确定特定活动事件或流程事件在一段时间内的发生频率。

第 329 页的『[时间处理](#)』

在报告中，请考虑业务流程编排器观察器处理时间戳记和持续时间的方式。

相关任务

第 339 页的『[示例：使用用户定义报告](#)』

本场景提供一个示例，阐述如何使用用户定义报告。

相关参考

第 340 页的『报告属性』

使用属性来定义报告内容以及对结果进行过滤。可用的属性视报告类型而定。

第 342 页的『业务流程编排器观察器的业务流程事件』

如果您要求对 WebSphere Integration Developer 中的业务流程元素进行监视，那么将发送业务流程事件。在这些事件中，有一小部分事件可供业务流程编排器观察器使用。

第 343 页的『与性能相关的属性』

运行报告定义所需的时间会有所变化。为了提高生成报告时的性能，可以对报告定义进行优化。有一些通用规则可以帮助您评估报告属性对性能的影响。

示例：使用用户定义报告

本场景提供一个示例，阐述如何使用用户定义报告。

开始之前

您的工厂生产三种不同的产品 Item1、Item2 和 Item3。制造和交货流程作为 WebSphere Process Server 中的 SOA 流程建模和运行。每个客户订单都由相应流程模板的专用流程实例表示。将产品交付客户后，交付流程将进入最终状态“已完成”。如果客户取消订单，那么相应的流程实例将终止并进入“已终止”状态。

其中一位取消了订单的客户投诉响应时间过长。您想要了解处理此订单耗时过长的原因。

关于此任务

创建用户定义报告，查找处于“已终止”状态并且工作时间超过两天的流程实例。此外，报告应该揭示已终止的流程实例的问题。

过程

1. 检索属于客户订单的流程实例数据。

客户名、地址以及订单号是业务数据的组成部分，因此包含在流程消息中。但是，由于业务对象不是公共事件基础结构（CEI）事件的组成部分，因此业务流程编排器观察器无法使用该对象的内容。但是，您知道要查找处于“已终止”状态并且工作时间超过两天的流程实例。

- a. 在导航栏的**流程报告**部分中，选择**创建新报告**。
 - b. 由于您关注流程实例的状态，因此，请选择报告类型**快照报告**。
 - c. 在选择快照类型页面上，选择**在特定日期和时间获取快照**。指定紧跟在订单取消操作后的日期和时间作为合格的快照日期。
 - d. 在报告内容页面上，在报告内容中添加**流程实例标识**、**流程工作时间**、**流程已开始**和**流程已完成**。
 - e. 在过滤器内容页面上，指定**流程工作时间大于 2 天**和**流程状态等于“已终止”**作为过滤器内容，然后运行该报告。
 - f. 在报告结果页面上，检查流程实例标识、开始日期和完成日期，以查找与客户订单相对应的流程实例。如果报告结果与您的预期不符，例如，显示的流程实例过多，那么请单击**编辑**以修改搜索条件。
 - g. 由于步骤 2 需要使用流程实例标识，因此请将该标识复制到剪贴板。
- #### 2. 获取用于揭示特定流程实例的问题的信息。

- a. 在导航栏的**流程报告**部分中，选择**创建新报告**。
- b. 选择报告类型**快照报告**。

请不要使用“时间段报告”类型。您感兴趣的是与快照报告相关的属性。要查看差别，请定义并运行属性完全相同的时间段报告。

- c. 在选择快照类型页面上，选择**在特定日期和时间获取快照**。指定紧跟在订单取消操作后的日期和时间作为合格的快照日期。
 - d. 在报告内容页面上，在报告内容中添加**流程实例标识**、**活动名称**、**活动启动时间**和**流程完成时间**。
 - e. 在过滤器内容页面上，指定**流程实例标识等于 *your_customer's_process_instance_ID***作为过滤器内容，然后运行该报告。该报告将揭示耗用最多时间的活动。
 - f. 可选：如果您需要进一步的信息以确定延迟问题的根本原因，请编辑并再次运行该报告。
 - g. 保存报告定义。
3. 最后，您希望将来避免再发生此类情况。在每个工作日结束时，您希望有一个报告列示所有由于资源限制或故障而有超出时间限制危险的活动订单流程。
 - a. 编辑已保存的报告定义。在选择快照类型页面上，将快照类型更改为**立即获取快照**，删除过滤器内容**流程实例标识等于 *your_customer's_process_instance_ID***，然后添加表达式**流程工作时间大于 1 天**。
 - b. 运行修改后的报告，并检查是否不存在符合新过滤器条件的流程实例。
 - c. 保存该报告，以便可以在每个工作日结束时运行该报告。

相关任务

第 334 页的『创建用户定义报告』

与预定义的列表和图表相比，用户定义的流程报告和活动报告更为灵活。此外，您可以存储和重复使用报告定义，并可以导出报告结果。

第 334 页的『创建用户定义的快照报告』

您可以定义用户定义报告，以便获取所指定日期和时间的状态信息快照。您还可以创建包含报告时间段内定期时间点（例如，每个月第一天的午夜）的状态快照的报告。

第 337 页的『创建用户定义的时间段报告』

您可以为一段时间内发生的流程事件或活动事件创建用户定义报告。还可以创建根据报告周期涵盖多个时间段的报告。

报告属性

使用属性来定义报告内容以及对结果进行过滤。可用的属性视报告类型而定。

每个定义为报告内容的属性都是该报告中某个列的名称。此外，通过使用属性，可以对查询结果进行过滤。还可以定义报告中未包括的属性的过滤器条件。

属性	描述	快照报告	时间段报告
活动已完成	活动实例进入下列其中一种结束状态的时间：已失败、已完成、已被跳过、已终止或已到期。	X	X

属性	描述	快照报告	时间段报告
活动事件	活动事件的事件码。	X	X
活动事件数	活动实例发出的活动事件数。	X	X
活动实例标识	活动实例标识。	X	X
活动种类	活动实例的种类。	X	X
活动的上一个用户名	上一个对此活动启动操作的用户的名称。	X	X
活动名称	活动实例的名称。	X	X
活动启动时间	活动实例的启动时间。	X	X
活动状态	活动实例在该事件后所处的状态。	X	X
活动模板标识	活动模板标识。	X	X
活动的平均持续时间	所有活动实例的平均持续时间，以秒计。	X	X
流程的平均持续时间	所有流程实例的平均持续时间，以秒计。	X	X
事件时间	事件的发生时间。	X	X
异常文本	如果活动事件由异常触发，那么异常消息可以是事件数据的组成部分并存储在此字段中。	X	X
处于某个状态的活动数	处于指定状态的活动实例的数目。	X	
活动事件数	在指定时间段内发生的活动事件的数目。		X
流程事件数	在指定时间段内发生的流程事件的数目。		X
处于某个状态的流程数	处于指定状态的流程实例的数目。	X	
流程活动数	流程实例的至少发出了一个事件的活动数。	X	X
流程活动事件数	属于某个流程实例的活动事件的数目。	X	X
流程完成时间	流程实例进入下列其中一种结束状态的时间：已补偿、补偿失败、已失败、已完成或已终止。	X	X
流程删除时间	从业务流程编排器数据库中删除流程的时间。	X	X
流程事件	流程实例事件的事件码。	X	X
流程事件数	流程实例发出的流程事件的数目。	X	X
流程实例标识	流程实例标识。	X	X
流程的上个用户名	上一个对此流程启动操作的用户的名称。	X	X
流程启动时间	流程实例的启动时间。	X	X
流程状态	流程实例在该事件后所处的状态。	X	X
流程模板标识	流程模板标识。	X	X
流程模板名称	与流程实例相关联的流程模板。	X	X
流程工作时间	流程实例的持续时间。此值是该流程中包含的所有基本活动的工作时间之和。基本活动是未包含其他活动的无结构活动。	X	X
快照编号	在具有报告周期的快照报告中，此属性标识报告周期中的特定快照。	X	
时间片编号	在具有报告周期的时间段报告中，此属性标识报告周期中的特定时间片。		X
用户名	与事件相关联的用户的用户标识。	X	X
生效时间	流程模板的生效时间。	X	X

业务流程编排器观察器的业务流程事件

如果您要求对 WebSphere Integration Developer 中的业务流程元素进行监视，那么将发送业务流程事件。在这些事件中，有一小部分事件可供业务流程编排器观察器使用。

业务流程可能会引起下列类型的事件：

- 『流程事件』
- 『活动事件』

根据 WebSphere Integration Developer 中设置的不同，可能会发生 6.0.2 事件和 6.1 事件。

业务流程编排器观察器不要求事件包含业务数据。

流程事件

下表对所有可以通过业务流程编排器观察器报告的流程事件进行描述。

代码	描述
21000	流程已启动
21001	流程已暂挂
21002	流程已恢复
21004	流程已完成
21005	流程已终止
21019	流程已重新启动
42001	流程已失败
42003	流程正在进行补偿
42004	流程已进行补偿
42046	流程补偿失败
42009	流程正在终止
42010	流程即将失败

活动事件

下表对所有可以通过业务流程编排器观察器报告的活动事件进行描述。

代码	描述
21006	活动已就绪
21007	活动已启动
21011	活动已完成
21021	声明已取消
21022	活动已声明
21027	活动已终止
21080	活动已失败
21081	活动已到期
42005	活动已被跳过
42015	活动已停止

代码	描述
42031	活动已强制再次执行
42032	活动已强制完成
42036	活动接收到消息

相关参考

第 525 页的『业务流程事件』

如果您要求对 WebSphere Integration Developer 中的业务流程元素进行监视，那么将发送业务流程事件。下面列示了业务流程可以发出的所有事件。

与性能相关的属性

运行报告定义所需的时间会有所变化。为了提高生成报告时的性能，可以对报告定义进行优化。有一些通用规则可以帮助您评估报告属性对性能的影响。

指定过滤器

使用适当的过滤器来限制所检索的数据量。请考虑按日期或者活动实例或流程实例的其他属性对报告结果进行限制。对于快照报告而言，请将 ReportAtSnapshotRange 配置参数设置为适当的值。

时间段报告与快照报告

与时间段报告相比，快照报告更容易使性能下降。

具有报告周期的报告

定义了报告周期的报告容易使性能下降，为查询定义了许多时间段或快照时尤其如此。

聚集

诸如事件总数或者实例平均持续时间之类的聚集会导致处理大量数据，从而使性能下降。

显示的结果数

如果您只对报告中的某些结果感兴趣，那么请指定阈值，以便对结果中的条目数进行限制。这将减少数据库与用户界面之间传输的数据量。

但是，如果定义了排序顺序，那么必须从数据库中收集所有结果数据，然后才能对数据进行排序。在这种情况下，减少所显示的结果数并不会提高性能。而是，您应该设置适当的过滤器表达式。

事件信息和实例信息

在观察器数据库中，与事件相关的信息存储在事件数据库表中，而与活动实例和流程实例相关的信息存储在实例数据库表中。如果创建的报告同时包含与实例相关的信息和特定于事件的信息，那么将对这些表进行连接以获取所需的信息。如果创建的报告只包含一种信息，那么不会对这些表进行连接。因此，只包含一种信息的报告的性能通常优于同时查询实例相关信息和事件特定信息的报告。

相关参考

第 234 页的『更改业务流程编排器观察器的配置参数』

对于启用验证功能以及提高性能而言，调整业务流程编排器观察器和事件收集器应用程序的配置参数至关重要。

使用已保存的用户定义报告定义

如果已保存了报告定义，那么可以根据需要运行报告、编辑报告定义或者使用该报告定义的副本来创建类似的报告。此外，还可以通过异步方式运行报告并导出报告结果。

运行已保存的用户定义报告定义

您可以在需要时运行已保存的报告定义。如果报告包含参数，那么可以在每次运行该报告时设置所需的参数值。

过程

1. 要运行已保存的报告定义，请在导航窗格中单击报告名。
 - 如果报告定义未包含任何参数，那么将显示所生成的报告。
 - 如果报告定义包含参数，那么将显示运行报告页面。您可以更改参数值，然后单击**运行**。

将显示所生成的报告。



2. 可选： 导出报告结果。

要以 CSV 格式导出所报告的数据，请单击**导出**。选择是要打开所生成的导出数据还是要将其保存在硬盘上，然后单击**确定**。

以异步方式运行已保存的用户定义报告定义



运行查询时，可以通过异步方式运行已保存的报告，以便继续完成工作。

关于此任务

要以异步方式运行已保存的报告定义，请单击**显示弹出菜单**图标 ()，然后单击“**异步搜索**”图标 ()。

过程

如果报告定义包含参数，那么将显示“运行报告”页面。您可以更改参数值，然后单击“**运行**”。

- 异步搜索成功完成后，在导航窗格中将显示“异步搜索已完成”图标 ()。然后，请单击报告的名称以查看搜索结果。
- 如果异步搜索未能成功完成，那么将显示“异步搜索已失败”图标 ()。



使用弹出菜单导出报告结果

对于已保存的用户定义报告，可以导出报告结果以便进行后续外部处理，而无需运行该报告。

关于此任务

此选项仅适用于已保存并且未包含参数的用户定义报告定义。

过程

1. 要导出已保存的报告定义的报告结果，请单击**显示弹出菜单**图标 ()，然后单击“导出”图标 ()。
2. 选择是要打开还是要保存所生成的导出数据，然后单击**确定**。所报告的数据将被导出。

使用导出客户机导出报告结果

对于已保存的用户定义报告，可以使用导出客户机命令行工具来运行那些报告并导出报告结果以便进行后续外部处理。

开始之前

此选项仅适用于已保存并且未包含参数的用户定义报告定义。

导出客户机工具 `wps_install_root/ProcessChoreographer/client/exportclient.jar` 必须安装在本地工作站上。

过程

要运行报告并导出报告结果，请使用命令行来启动导出客户机。

在 Windows 平台上，输入以下命令：`java -jar exportclient.jaroptions`

在 Linux、UNIX 和 iOS 平台上，输入以下命令：`java -jar exportclient.jaroptions`

可以直接在命令行上以 `-option value` 格式指定选项，也可以指定属性文件的名称。这些选项的格式为 `option=value`。命令行上指定的选项优先于属性文件中指定的选项。

下列选项有效：

表 15. 导出客户机的有效选项



选项	描述
help	显示用法信息。
verbose	在导出结果时显示其他信息，这些信息有助于进行调试。
unicode	以 UTF-8 编码格式导出结果。缺省值是本地操作系统编码。
o	覆盖任何现有文件。在缺省情况下，如果该文件已存在，那么将出错。
properties	此选项定义包含其他选项的文件的标准名称。
url	运行中的业务流程编排器观察器的完整 URL。缺省值是 <code>http://localhost:9080</code> 。
out	此选项定义用于存储导出结果的文件的名称。缺省值是 <code>report name.csv</code> 。
userid	如果已启用安全性，那么必须指定有效的用户标识。
password	如果已启用安全性，那么必须指定有效的密码。
reportname	必须指定已保存的报告定义的名称。

编辑和复制已保存的用户定义报告定义

您可以更改已保存的报告定义的设置，也可以使用该报告定义的副本来创建类似的报告。

过程

1. 单击**显示弹出菜单**图标 ()，然后执行下列其中一项操作：

- 要编辑报告定义，请单击**修改**图标 ()。
- 要复制报告定义，请单击**复制**图标 ()。

将打开总结页面。此页面显示了该报告的时间设置、报告内容及其过滤器设置。

请单击每个总结部分下面的链接，以更改相应的设置。无法更改报告类型。

2. 可选： 要编辑时间设置，请单击**修改报告的日期和报告周期设置**。

根据所定义报告类型的不同，将打开选择快照类型页面或选择时间段类型页面。

3. 可选： 要修改报告内容，请单击**修改结果内容**。

将打开指定报告内容页面。

对于具有报告周期的报告，属性列表将包含快照编号属性或时间片编号属性，这取决于所定义报告的类型。您无法删除此属性。

4. 可选： 要修改过滤器设置，请单击**修改过滤器设置**。

将打开指定过滤器内容页面。

5. 在总结页面上，执行下列其中一项操作：

- 如果报告定义未包含参数，那么请单击**运行**。

将显示所生成的报告。

- 如果报告定义包含参数，那么请单击**下一步**。

您可以更改参数值，然后单击**运行**。将显示所生成的报告。

如果报告结果与您的预期不符，那么可以单击**编辑**以更改报告的设置。

6. 在报告结果页面上，单击**保存**。如果您将要创建报告定义的副本，请输入新报告的名称，然后再次单击**保存**。

新报告将显示在导航窗格中。

相关概念

第 329 页的『时间处理』

在报告中，请考虑业务流程编排器观察器处理时间戳记和持续时间的方式。

相关参考

第 340 页的『报告属性』

使用属性来定义报告内容以及对结果进行过滤。可用的属性视报告类型而定。

第 342 页的『业务流程编排器观察器的业务流程事件』

如果您要求对 WebSphere Integration Developer 中的业务流程元素进行监视，那么将发送业务流程事件。在这些事件中，有一小部分事件可供业务流程编排器观察器使用。

第 343 页的『与性能相关的属性』

运行报告定义所需的时间会有所变化。为了提高生成报告时的性能，可以对报告定义进行优化。有一些通用规则可以帮助您评估报告属性对性能的影响。



删除已保存的用户定义报告定义

为了保持导航窗格整洁而易于管理，您应该删除过期的以及冗余的报告定义。

开始之前

无法恢复已删除的报告定义。

过程

要删除报告定义，请单击显示弹出菜单图标 ()，然后单击删除图标 ()。

结果

该报告名称将从导航窗格中消失。

第 4 部分 开发和部署模块

第 10 章 为业务流程和任务开发客户机应用程序

可以使用建模工具来构建和部署业务流程和任务。在运行时，例如在启动流程或者声明和完成任务时，这些流程和任务将进行交互。可以使用业务流程编排器资源管理器与流程和任务进行交互，也可以使用业务流程编排器 API 来开发用于执行这些交互的定制应用程序。

关于此任务

这些客户机可以是使用了业务流程编排器资源管理器 JavaServer Faces (JSF) 组件的 Enterprise JavaBeans™ (EJB) 客户机、Web Service 客户机或 Web 客户端。业务流程编排器提供了用于 Web Service 的 Enterprise JavaBeans (EJB) API 和接口，以使您能够开发这些客户机。EJB API 可以由包括另一个 EJB 应用程序在内的任何 Java 应用程序访问。可以从 Java 环境或 Microsoft .Net 环境中访问 Web Service 的接口。

相关概念

第 15 页的『业务流程的调用方案』

业务流程是 SCA (服务组件体系结构) 组件实现类型。它可以发布服务供其他伙伴使用，也可以使用由其他伙伴提供的服务。业务流程可以通过业务流程编排器 API 提供的服务提供程序、其他 SCA 服务组件的 SCA 服务提供程序或者调用了其他 SCA 服务组件 (包括其他业务流程) 的 SCA 客户机。

第 11 章 为业务流程和人员任务开发 EJB 客户机应用程序

EJB API 提供了一组用于开发 EJB 客户机应用程序的通用方法，这些方法使您能够处理 WebSphere Process Server 上安装的业务流程和人员任务。

关于此任务

使用这些 Enterprise JavaBeans (EJB) API，可以创建客户机应用程序来执行下列操作：

- 管理流程和任务的生命周期，从启动它们直至在它们完成时将其删除
- 修复活动和流程
- 在工作组成员之间管理和分配工作负载

EJB API 作为下面这两个无状态会话企业 Bean 提供：

- `BusinessFlowManagerService` 接口为业务流程应用程序提供方法
- `HumanTaskManagerService` 接口为基于任务的应用程序提供方法

有关 EJB API 的更多信息，请参阅 `com.ibm.bpe.api` 包和 `com.ibm.task.api` 包中的 Javadoc。

下列步骤对开发 EJB 客户机应用程序时需要执行的操作进行概述。

过程

1. 确定应用程序要提供的功能。
2. 确定将要使用的会话 Bean。

根据应用程序的实现方案，可以使用其中一个会话 Bean，也可以同时使用这两个会话 Bean。

3. 确定应用程序的用户所需的权限。

必须对应用程序的用户指定适当的授权角色，使他们有权调用应用程序中包含的方法以及查看这些方法返回的对象和对象属性。在创建适当会话 Bean 的实例时，WebSphere Application Server 将使上下文与该实例相关联。上下文包含关于调用者主体标识、组成员资格列表和角色的信息。此信息用于检查调用者对每个调用的权限。

Javadoc 提供了每个方法的权限信息。

4. 确定应用程序的显示方式。

可以通过本地方式或远程方式调用 EJB API。

5. 开发应用程序。
 - a. 访问 EJB API。
 - b. 使用 EJB API 与流程或任务进行交互。
 - 查询数据。
 - 处理数据。

访问 EJB API

Enterprise JavaBeans (EJB) API 作为两个无状态会话企业 Bean 提供。业务流程应用程序和任务应用程序通过 Bean 的 Home 接口访问适当的会话企业 Bean。

关于此任务

`BusinessFlowManagerService` 接口为业务流程应用程序提供方法，`HumanTaskManagerService` 接口为基于任务的应用程序提供方法。此应用程序可以是任何 Java 程序，也可以是另一个 Enterprise JavaBeans (EJB) 应用程序。

访问会话 Bean 的远程接口

EJB 客户机应用程序通过会话 Bean 的远程 Home 接口来访问该 Bean 的远程接口。

关于此任务

会话 Bean 可以是用于流程应用程序的 `BusinessFlowManager` 会话 Bean，也可以是用于任务应用程序的 `HumanTaskManager` 会话 Bean。

过程

1. 在应用程序部署描述符中添加对会话 Bean 的远程接口的引用。请添加对下列其中一个文件的引用：
 - `application-client.xml` 文件（用于 Java 2 Platform, Enterprise Edition (J2EE) 客户机应用程序）
 - `web.xml` 文件（用于 Web 应用程序）
 - `ejb-jar.xml` 文件（用于 Enterprise JavaBeans (EJB) 应用程序）

以下示例说明如何在流程应用程序中引用远程 Home 接口：

```
<ejb-ref>
  <ejb-ref-name>ejb/BusinessFlowManagerHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <home>com.ibm.bpe.api.BusinessFlowManagerHome</home>
  <remote>com.ibm.bpe.api.BusinessFlowManager</remote>
</ejb-ref>
```

以下示例说明如何在任务应用程序中引用远程 Home 接口：

```
<ejb-ref>
  <ejb-ref-name>ejb/HumanTaskManagerHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <home>com.ibm.task.api.HumanTaskManagerHome</home>
  <remote>com.ibm.task.api.HumanTaskManager</remote>
</ejb-ref>
```

如果使用 WebSphere Integration Developer 在部署描述符中添加 EJB 引用，那么部署该应用程序时将自动为 EJB 引用创建绑定。有关添加 EJB 引用的更多信息，请参阅 WebSphere Integration Developer 文档。

2. 将生成的存根与应用程序打包到一起。

如果该应用程序与 `BPEContainer` 应用程序或 `TaskContainer` 应用程序在不同的 Java 虚拟机 (JVM) 中运行，那么完成下列操作：

- a. 对于流程应用程序而言，请将 `<install_root>/ProcessChoreographer/client/bpe137650.jar` 文件与应用程序的企业归档 (EAR) 文件打包到一起。

- b. 对于任务应用程序而言, 请将 `<install_root>/ProcessChoreographer/client/task137650.jar` 文件与应用程序的 EAR 文件打包到一起。
- c. 将应用程序模块清单文件中的 **Classpath** 参数设置为包括该 JAR 文件。

该应用程序模块可以是 J2EE 应用程序、Web 应用程序或 EJB 应用程序。

- d. 如果在业务流程或人员任务中使用了复杂数据类型, 并且客户机不在 EJB 应用程序或 Web 应用程序中运行, 那么必须将相应的 XSD 或 WSDL 文件与应用程序的 EAR 文件打包到一起。
3. 通过 Java 命名和目录接口 (JNDI) 来定位会话 Bean 的远程 Home 接口。

以下示例说明流程应用程序中的此步骤:

```
// Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

// Lookup the remote home interface of the BusinessFlowManager bean
Object result =
    initialContext.lookup("java:comp/env/ejb/BusinessFlowManagerHome");

// Convert the lookup result to the proper type
BusinessFlowManagerHome processHome =
    (BusinessFlowManagerHome)javax.rmi.PortableRemoteObject.narrow
    (result,BusinessFlowManagerHome.class);
```

会话 Bean 的远程 Home 接口包含 EJB 对象的 create 方法。该方法返回该会话 Bean 的远程接口。

4. 访问会话 Bean 的远程接口。

以下示例说明流程应用程序中的此步骤:

```
BusinessFlowManager process = processHome.create();
```

对会话 Bean 的访问权并不保证调用者能够执行该 Bean 所提供的所有操作; 该调用者还必须有权执行这些操作。在创建该会话 Bean 的实例时, 将使上下文与该会话 Bean 的实例相关联。此上下文包含调用者的主体标识和组成员资格列表, 并指示了调用者是否具有某个业务流程编排器 J2EE 角色。每次执行调用时, 将使用上下文来检查调用者的权限, 即使未启用全局安全性亦如此。如果未启用全局安全性, 那么调用者的主体标识值为 UNAUTHENTICATED。

5. 调用服务接口发布的业务函数。

以下示例说明流程应用程序中的此步骤:

```
process.initiate("MyProcessModel",input);
```

应用程序中执行的调用作为事务运行, 而事务以下列其中一种方式建立和结束:

- 由 WebSphere Application Server 自动建立和结束 (部署描述符指定了 TX_REQUIRED)。
- 由应用程序明确建立和结束。可以将多个应用程序调用捆绑成一个事务:

```
// Obtain user transaction interface
UserTransaction transaction=
    (UserTransaction)initialContext.lookup("jta/usertransaction");

// Begin a transaction
transaction.begin();
```

```
// Applications calls ...

// On successful return, commit the transaction
transaction.commit();
```

提示: 为了防止数据库锁定冲突, 请避免并行运行与下列内容相似的语句:

```
// Obtain user transaction interface
UserTransaction transaction=
    (UserTransaction)initialContext.lookup("jta/usertransaction");

transaction.begin();

//read the activity instance
process.getActivityInstance(aiid);
//claim the activity instance
process.claim(aiid);

transaction.commit();
```

`getActivityInstance` 方法和其他读操作将设置读锁。在本示例中, 活动实例上的读锁将升级为活动实例上的更新锁。当这些事务并行运行时, 这会导致数据库死锁。

示例

以下是在任务应用程序中执行步骤 3 到 5 的示例。

```
//Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

//Lookup the remote home interface of the HumanTaskManager bean
Object result =
    initialContext.lookup("java:comp/env/ejb/HumanTaskManagerHome");

//Convert the lookup result to the proper type
HumanTaskManagerHome taskHome =
    (HumanTaskManagerHome)javax.rmi.PortableRemoteObject.narrow
    (result,HumanTaskManagerHome.class);

...
//Access the remote interface of the session bean.
HumanTaskManager task = taskHome.create();

...
//Call the business functions exposed by the service interface
task.callTask(tkid,input);
```

访问会话 Bean 的本地接口

EJB 客户机应用程序通过会话 Bean 的本地 Home 接口来访问该 Bean 的本地接口。

关于此任务

会话 Bean 可以是用于流程应用程序的 `BusinessFlowManager` 会话 Bean, 也可以是用于人员任务应用程序的 `HumanTaskManager` 会话 Bean。

过程

1. 在应用程序部署描述符中添加对会话 Bean 的本地接口的引用。请添加对下列其中一个文件的引用:
 - `application-client.xml` 文件 (用于 Java 2 Platform, Enterprise Edition (J2EE) 客户机应用程序)

- web.xml 文件（用于 Web 应用程序）
- ejb-jar.xml 文件（用于 Enterprise JavaBeans (EJB) 应用程序）

以下示例说明如何在流程应用程序中引用本地 Home 接口：

```
<ejb-local-ref>
  <ejb-ref-name>ejb/LocalBusinessFlowManagerHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <local-home>com.ibm.bpe.api.LocalBusinessFlowManagerHome</local-home>
  <local>com.ibm.bpe.api.LocalBusinessFlowManager</local>
</ejb-local-ref>
```

以下示例说明如何在任务应用程序中引用本地 Home 接口：

```
<ejb-local-ref>
  <ejb-ref-name>ejb/LocalHumanTaskManagerHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <local-home>com.ibm.task.api.LocalHumanTaskManagerHome</local-home>
  <local>com.ibm.task.api.LocalHumanTaskManager</local>
</ejb-local-ref>
```

如果使用 WebSphere Integration Developer 在部署描述符中添加 EJB 引用，那么部署该应用程序时将自动为 EJB 引用创建绑定。有关添加 EJB 引用的更多信息，请参阅 WebSphere Integration Developer 文档。

2. 通过 Java 命名和目录接口 (JNDI) 来定位会话 Bean 的本地 Home 接口。

以下示例说明流程应用程序中的此步骤：

```
// Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

// Lookup the local home interface of the BusinessFlowManager bean

LocalBusinessFlowManagerHome processHome =
    (LocalBusinessFlowManagerHome)initialContext.lookup
    ("java:comp/env/ejb/LocalBusinessFlowManagerHome");
```

会话 Bean 的本地 Home 接口包含 EJB 对象的 create 方法。该方法返回该会话 Bean 的本地接口。

3. 访问会话 Bean 的本地接口。

以下示例说明流程应用程序中的此步骤：

```
LocalBusinessFlowManager process = processHome.create();
```

对会话 Bean 的访问权并不保证调用者能够执行该 Bean 所提供的所有操作；调用者还必须有权执行这些操作。在创建该会话 Bean 的实例时，将使上下文与该会话 Bean 的实例相关联。此上下文包含调用者的主体标识和组成员资格列表，并指示了调用者是否具有某个业务流程编排器 J2EE 角色。每次执行调用时，将使用上下文来检查调用者的权限，即使未启用全局安全性亦如此。如果未启用全局安全性，那么调用者的主体标识值为 UNAUTHENTICATED。

4. 调用服务接口发布的业务函数。

以下示例说明流程应用程序中的此步骤：

```
process.initiate("MyProcessModel",input);
```

应用程序中执行的调用作为事务运行，而事务以下列其中一种方式建立和结束：

- 由 WebSphere Application Server 自动建立和结束（部署描述符指定了 TX_REQUIRED）。
- 由应用程序明确建立和结束。可以将多个应用程序调用捆绑成一个事务：

```
// Obtain user transaction interface
UserTransaction transaction=
    (UserTransaction)initialContext.lookup("jta/usertransaction");

// Begin a transaction
transaction.begin();

// Applications calls ...

// On successful return, commit the transaction
transaction.commit();
```

提示： 为了防止数据库死锁，请避免并行运行与下列内容相似的语句：

```
// Obtain user transaction interface
UserTransaction transaction=
    (UserTransaction)initialContext.lookup("jta/usertransaction");

transaction.begin();

//read the activity instance
process.getActivityInstance(aiid);
//claim the activity instance
process.claim(aiid);

transaction.commit();
```

getActivityInstance 方法和其他读操作将设置读锁。在本示例中，活动实例上的读锁将升级为活动实例上的更新锁。当这些事务并行运行时，这会导致数据库死锁。

示例

以下是在任务应用程序中执行步骤 2 到 4 的示例。

```
//Obtain the default initial JNDI context
InitialContext initialContext = new InitialContext();

//Lookup the local home interface of the HumanTaskManager bean
LocalHumanTaskManagerHome taskHome =
    (LocalHumanTaskManagerHome)initialContext.lookup
    ("java:comp/env/ejb/LocalHumanTaskManagerHome");

...
//Access the local interface of the session bean
LocalHumanTaskManager task = taskHome.create();

...
//Call the business functions exposed by the service interface
task.callTask(tkIID,input);
```

查询与业务流程和任务相关的对象

客户机应用程序处理与业务流程和任务相关的对象。可以查询数据库中与业务流程和任务相关的对象，以检索这些对象的特定属性。

关于此任务

配置业务流程编排器期间，将使一个关系数据库与业务流程容器和任务容器相关联。此数据库存储所有用于管理业务流程和任务的模板（模型）和实例（运行时）数据。您使用类似于 SQL 的语法来查询此数据。

可以执行一次性的查询来检索某个对象的特定属性。也可以将常用查询保存下来并将存储的这些查询包括在应用程序中。

对业务流程以及与任务相关的对象执行的查询

使用服务 API 的 query 方法或 queryAll 方法来检索已存储的业务流程信息和任务信息。

query 方法可以由所有用户调用，它返回工作项的相关对象的属性。queryAll 方法只能由具有下列某个 J2EE 角色的用户调用：BPESystemAdministrator、TaskSystemAdministrator、BPESystemMonitor 或 TaskSystemMonitor。此方法返回数据库中存储的所有对象的属性。

所有 API 查询都映射到 SQL 查询。所生成的 SQL 查询的格式取决于下列因素：

- 该查询是否已被具有某个 J2EE 角色的用户调用。
- 所查询的对象。可以使用预定义的数据库视图来查询对象属性。
- 是否插入了 FROM 子句、连接条件以及用于进行访问控制的特定于用户的条件。

可以在查询中同时指定定制属性和变量属性。如果在查询中指定多个定制属性或变量属性，那么将导致对相应的数据库表进行自连接。根据数据库系统的不同，这些 query() 调用可能会影响性能。

还可以使用 createStoredQuery 方法将查询存储在业务流程编排器数据库中。在定义存储查询时，需要提供查询条件。将在运行存储查询时动态地应用此条件，即，数据在运行时进行组装。如果存储查询包含参数，那么运行查询时还将对这些参数进行解析。

有关业务流程编排器 API 的更多信息，请参阅 com.ibm.bpe.api 包（包含与流程相关的方法）和 com.ibm.task.api 包（包含与任务相关的方法）中的 Javadoc。

API query 方法的语法

业务流程编排器 API 查询语法与 SQL 查询类似。查询可以包含 SELECT 子句、WHERE 子句、Order-by 子句、Skip-tuples 参数、threshold 参数和 timezone 参数。

查询语法取决于对象类型。下表列示了用于每种对象类型的语法。

表 16.

对象	语法
流程模板	<pre>ProcessTemplateData[] queryProcessTemplates (java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer threshold, java.util.TimeZone timezone);</pre>
任务模板	<pre>TaskTemplate[] queryTaskTemplates (java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer threshold, java.util.TimeZone timezone);</pre>

表 16. (续)

对象	语法
与业务流程和任务相关的数据	<pre>QueryResultSet query (java.lang.String selectClause, java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer skipTuples java.lang.Integer threshold, java.util.TimeZone timezone);</pre>

Select 子句:

查询函数中的 Select 子句标识查询所要返回的对象属性。

Select 子句描述查询结果。它指定了一组名称，这些名称标识了要返回的对象属性，即结果列。Select 子句语法与 SQL 中 SELECT 子句的语法相同；请使用逗号来分隔此子句的各个组成部分。此子句的每个部分都必须指定某个预定义视图中的列。必须按视图名和列名对各个列进行完全限定。QueryResultSet 对象返回列的顺序与 Select 子句中指定列的顺序相同。

Select 子句不支持诸如 AVG()、SUM()、MIN() 或 MAX() 之类的 SQL 聚集函数。

要选择多个“名称/值”对的属性，例如定制属性以及可查询变量的属性，请在视图名中添加 1 位数的计数器。此计数器值可以是 1 到 9。

Select 子句示例

- "WORK_ITEM.OBJECT_TYPE, WORK_ITEM.REASON"

获取相关对象的对象类型以及工作项的分配原因。

- "DISTINCT WORK_ITEM.OBJECT_ID"

获取调用者具有其工作项的相应对象的所有标识（排除重复项）。

- "ACTIVITY.TEMPLATE_NAME, WORK_ITEM.REASON"

获取调用者具有其工作项的相应活动的名称及其分配原因。

- "ACTIVITY.STATE, PROCESS_INSTANCE.STARTER"

获取活动的状态及其相关流程实例的启动者。

- "DISTINCT TASK.TKIID, TASK.NAME"

获取调用者具有其工作项的相应任务的所有标识和名称（排除重复项）。

- "TASK_CPROP1.STRING_VALUE, TASK_CPROP2.STRING_VALUE"

获取 Where 子句中进一步指定的定制属性的值。

- "QUERY_PROPERTY1.STRING_VALUE, QUERY_PROPERTY2.INT_VALUE"

获取可查询变量的属性值。WHERE 子句进一步指定这些部件。

- "COUNT(DISTINCT TASK.TKIID)"

返回满足 Where 子句的唯一任务的工作项数。

Where 子句:

查询函数中的 Where 子句描述要对查询范围应用的过滤器条件。

WHERE 子句的语法与 SQL 的 WHERE 子句语法类似。不需要明确地对 API Where 子句添加 SQL FROM 子句或 JOIN 谓词，在运行查询时将自动添加这些构造。如果您不想应用过滤器条件，那么必须将 Where 子句指定为 null。

Where 子句语法支持：

- 关键字：AND, OR, NOT
- 比较运算符：=、<=、<、<>、>、>= 和 LIKE

LIKE 操作支持对所查询的数据库定义的通配符。

- 集合操作：IN

下列规则也适用：

- 将对象标识常量指定为 ID('string-rep-of-oid')。
- 将二进制常量指定为 BIN('UTF-8 string')。
- 使用符号常量来代替整数枚举。例如，指定 ACTIVITY.STATE=ACTIVITY.STATE.STATE_READY 来代替活动状态表达式 ACTIVITY.STATE=2。
- 如果比较语句中的属性值包含单引号 (')，那么请成倍地指定单引号，例如 "TASK_CPROP.STRING_VALUE='d''automatisation'"。
- 要引用多个“名称/值”对的属性（例如定制属性），请对视图名添加由 1 位数字组成的后缀。例如："TASK_CPROP1.NAME='prop1' AND "TASK_CPROP2.NAME='prop2'"
- 将时间戳记常量指定为 TS('yyyy-mm-ddThh:mm:ss')。要引用当前日期，请指定 CURRENT_DATE 作为时间戳记。

在时间戳记中必须至少指定日期值或时间值：

- 如果仅指定日期，那么时间值将设置为零。
- 如果仅指定时间，那么日期将设置为当前日期。
- 如果指定了日期，那么必须指定 4 位的年份；月份值和天值是可选的。缺少的月份值和天值将设置为 01。例如，TS('2003') 与 TS('2003-01-01T00:00:00') 相同。
- 如果指定了时间，那么这些值使用 24 小时制表达。例如，如果当前日期是 2003 年 1 月 1 日，那么 TS('T16:04') 或 TS('16:04') 与 TS('2003-01-01T16:04:00') 相同。

Where 子句示例

- 将对象标识与现有标识作比较

```
"WORK_ITEM.WIID = ID('_WI:800c00ed.df8d7e7c.feffff80.38')"
```

通常，此类 Where 子句通过上一个调用返回的现有对象标识以动态方式创建。如果此对象标识存储在 *wiid1* 变量中，那么可以将此子句构造为：

```
"WORK_ITEM.WIID = ID('" + wiid1.toString() + "')
```

- 使用时间戳记

```
"ACTIVITY.STARTED >= TS('2002-06-1T16.00.00')"
```

- 使用符号常量

```
"WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_OWNER"
```

- 使用布尔值 true 和 false

```
"ACTIVITY.BUSINESS_RELEVANCE = TRUE"
```

- 使用定制属性

```
"TASK_CPROP1.NAME = 'prop1' AND " TASK_CPROP1.STRING_VALUE = 'v1' AND  
TASK_CPROP2.NAME = 'prop2' AND " TASK_CPROP2.STRING_VALUE = 'v2'"
```

Order-by 子句:

查询函数中的 Order-By 子句指定查询结果集的排序条件。

您可以指定视图中的一组列，结果将按这些列进行排序。必须使用视图名和列名对这些列进行完全限定。最好指定 SELECT 子句中的列。

Order-by 子句语法与 SQL 中 ORDER BY 子句的语法相同；请使用逗号来分隔子句的各个组成部分。还可以指定 ASC 以便按升序对各个列进行排序，也可以指定 DESC 以便按降序对各个列进行排序。如果您不想对查询结果集进行排序，那么必须将 Order-by 子句指定为 null。

排序条件将在服务器上应用，即，使用服务器的语言环境来进行排序。如果指定了多个列，那么查询结果集将首先按第一个列的值进行排序，然后按第二个列的值进行排序，依此类推。在 Order-by 子句中，不能按位置指定列，这与 SQL 查询不同。

Order-by 子句示例

- "PROCESS_TEMPLATE.NAME"

按流程模板名称的字母顺序对查询结果进行排序。

- "PROCESS_INSTANCE.CREATED, PROCESS_INSTANCE.NAME DESC"

按创建日期对查询结果进行排序，并且，对于特定日期，按流程实例名的字母顺序逆序对结果进行排序。

- "ACTIVITY.OWNER, ACTIVITY.TEMPLATE_NAME, ACTIVITY.STATE"

依次按活动所有者、活动模板名和活动状态对查询结果进行排序。

Skip-tuples 参数:

Skip-tuples 参数指定在查询结果集开头要忽略（不返回给该查询结果集中的调用者）的查询结果集元组数。

通过将此参数与 threshold 参数配置使用，可以在客户机应用程序中实现分页，例如每次检索 20 项。

如果将此参数设置为 null，并且未设置 threshold 参数，那么将返回所有符合条件的元组。

Skip-tuples 参数示例

- new Integer(5)

指定不返回前 5 个符合条件的元组。

Threshold 参数:

查询函数中的 `Threshold` 参数对通过查询结果集从服务器返回到客户机的对象数进行限制。

由于生产方案中的查询结果集可以包含数千个甚至数百万个项，因此，最好始终指定阈值。`Threshold` 参数非常有用。例如，在每次只应该显示少量项目的图形用户界面中，就是这种情况。如果适当地设置 `Threshold` 参数，那么可以提高数据库查询的运行速度，并且可以减少需要从服务器传输到客户机的数据量。

如果将此参数设置为 `null`，并且未设置 `Skip-tuples` 参数，那么将返回所有符合条件的对象。

Threshold 参数示例

- `new Integer(50)`

指定返回 50 个符合条件的元组。

Timezone 参数:

查询函数中的 `Timezone` 参数定义该查询中的时间戳记常量的时区。

在启动查询的客户机与处理该查询的服务器之间，时区可能会不同。请使用 `Timezone` 参数指定 `Where` 子句中使用的时区，例如，使用该参数指定本地时间。查询结果集中返回的日期将采用该查询中指定的时区。

如果此参数设置为 `null`，那么将假定时间戳记常量是全球标准时间（UTC）。

Timezone 参数示例

- ```
process.query("ACTIVITY.AIID",
 "ACTIVITY.STARTED > TS('2005-01-01T17:40')",
 (String)null,
 (Integer)null,
 java.util.TimeZone.getDefault());
```

返回 2005 年 1 月 1 日本地时间 17:40 后启动的活动的对象标识。

- ```
process.query("ACTIVITY.AIID",
              "ACTIVITY.STARTED > TS('2005-01-01T17:40')",
              (String)null, (Integer)null, (TimeZone)null);
```

返回 2005 年 1 月 1 日 UTC 时间 17:40 后启动的活动的对象标识。例如，此时间比东部标准时间早 6 小时。

存储查询中的参数:

存储查询是指存储在数据库中并由名称标识的查询。限定元组在该查询运行时以动态方式进行组装。为了使存储查询可重复使用，可以在查询定义中使用将在运行时进行解析的参数。

例如，您已经定义了定制属性来存储客户名。可以定义查询来返回与特定客户 ACME Co. 相关联的任务。要查询此信息，查询中的 `WHERE` 子句可能与以下示例类似:

```
String whereClause =
    "TASK.STATE = TASK.STATE.STATE_READY
    AND WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER
    AND TASK_CPROP.NAME = 'company' AND TASK_CPROP.STRING_VALUE = 'ACME Co.'";
```

为了使此查询可重复使用，以便搜索客户 BCME Ltd，可以使用参数来存储定制属性的值。在任务查询中添加参数后，此查询可能与以下示例类似：

```
String whereClause =
    "TASK.STATE = TASK.STATE.STATE_READY
    AND WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER
    AND TASK_CPROP.NAME = 'company' AND TASK_CPROP.STRING_VALUE = '@param1'";
```

在运行时，将根据传递给 query 方法的参数列表对 @param1 参数进行解析。在查询中使用参数时，下列规则适用：

- 只能在 WHERE 子句中使用参数。
- 参数是字符串。
- 在运行时，将把参数替换为字符串。如果需要使用特殊字符，那么必须在 WHERE 子句中指定这些字符，或者在运行时作为参数组成部分传递这些字符。
- 包含字符串 @param 的参数名将与一个整数进行并置。最小数值为 1，此值指向在运行时传递给 query API 的参数列表中的第一项。
- 一个参数在 WHERE 子句中可以使用多次；该参数的所有出现将被替换为同一个值。

查询结果：

查询结果集包含查询结果。

结果集元素是满足调用者所指定 WHERE 子句并且调用者有权查看的对象属性。可以使用 API next 方法以相对方式读取元素，也可以使用 first 和 last 方法以绝对方式读取元素。由于查询结果集的隐式游标最初定位在第一个元素之前，所以，在读取元素前，必须调用 first 或 next 方法。可以使用 size 方法来确定集合中的元素数目。

查询结果集的元素由工作项的所选属性及其相关被引用对象（例如活动实例和流程实例）组成。QueryResultSet 元素的第一个属性（列）指定查询请求中 Select 子句所指定的第一个属性的值。QueryResultSet 元素的第二个属性（列）指定查询请求中 Select 子句所指定的第二个属性的值，依此类推。

可以通过调用与属性类型兼容的方法以及通过指定适当的列索引来检索属性值。列索引号从 1 开始。

属性类型	方法
字符串	getString
OID	getOID
时间戳记	getTimestamp getString getTimestampAsLong
整型	getInteger getShort getLong getString getBoolean

属性类型	方法
布尔值	getBoolean getShort getInteger getLong getString
byte[]	getBinary

示例:

运行以下查询:

```
QueryResultSet resultSet = process.query("ACTIVITY.STARTED,
ACTIVITY.TEMPLATE_NAME AS NAME,
WORK_ITEM.WIID, WORK_ITEM.REASON",
(String)null, (String)null,
(Integer)null, (TimeZone)null);
```

返回的查询结果集包含 4 列:

- 第 1 列是时间戳记
- 第 2 列是字符串
- 第 3 列是对象标识
- 第 4 列是整数

可以使用下列方法来检索属性值:

```
while (resultSet.next())
{
    java.util.Calendar activityStarted = resultSet.getTimestamp(1);
    String templateName = resultSet.getString(2);
    WIID wiid = (WIID) resultSet.getOID(3);
    Integer reason = resultSet.getInteger(4);
}
```

可以使用结果集的显示名, 例如, 在打印的表中, 将显示名用作标题。这些名称是视图的列名, 或者是查询中 AS 子句定义的名称。可以使用以下方法来检索本示例中的显示名:

```
resultSet.getColumnDisplayName(1) 返回 "STARTED"
resultSet.getColumnDisplayName(2) 返回 "NAME"
resultSet.getColumnDisplayName(3) 返回 "WIID"
resultSet.getColumnDisplayName(4) 返回 "REASON"
```

特定于用户的访问条件

根据 API 查询生成 SQL SELECT 语句时, 将添加特定于用户的访问条件。这些条件保证只将满足调用者所指定条件并且调用者有权使用的对象返回给调用者。

所添加的访问条件取决于该用户是否是系统管理员。

非系统管理员用户调用的查询

生成的 SQL WHERE 子句结合了 API WHERE 子句以及特定于用户的访问控制条件。该查询只检索用户有权访问的对象, 即, 用户拥有的工作项的相关对象。工作项表示将用户或用户组分配到业务对象 (例如任务或流程) 的授权角色。例如, 如果用户 John Smith 是给定任务的潜在所有者角色的成员, 那么将存在代表此关系的工作项对象。

例如，如果非系统管理员用户查询任务，并且组工作项未处于启用状态，那么将对 WHERE 子句添加以下访问条件：

```
FROM TASK TA, WORK_ITEM WI
WHERE WI.OBJECT_ID = TA.TKIID
AND ( WI.OWNER_ID = 'user'
      OR WI.OWNER_ID = null AND WI.EVERYBODY = true )
```

因此，如果 John Smith 要获取由他作为潜在所有者的任务的列表，那么 API WHERE 子句如下所示：

```
"WORK_ITEM.REASON == WORK_ITEM.REASON.REASON_POTENTIAL_OWNER"
```

这个 API WHERE 子句将使 SQL 语句包含以下访问条件：

```
FROM TASK TA, WORK_ITEM WI
WHERE WI.OBJECT_ID = TA.TKIID
AND ( WI.OWNER_ID = 'JohnSmith'
      OR WI.OWNER_ID = null AND WI.EVERYBODY = true)
AND WI.REASON = 1
```

这还意味着，如果 John Smith 要查看由他作为流程阅读者或流程管理员但他没有相应工作项的活动和任务，那么必须对查询的 SELECT、WHERE 或 Order-by 子句添加 PROCESS_INSTANCE 视图中的属性，例如 PROCESS_INSTANCE.PIID。

如果组工作项处于启用状态，那么将对 WHERE 子句添加其他访问条件，以允许用户访问该组有权访问的对象。

系统管理员调用的查询

系统管理员可以通过调用 query 方法来检索带有相关工作项的对象。在这种情况下，将在所生成的 SQL 查询中添加与 WORK_ITEM 视图的连接，但没有 WORK_ITEM.OWNER_ID 的访问控制条件。

在这种情况下，任务的 SQL 查询包含以下内容：

```
FROM TASK TA, WORK_ITEM WI
WHERE WI.OBJECT_ID = TA.TKIID
```

queryAll 查询

此类查询只能由系统管理员或系统监视员调用。既不会添加访问控制条件，也不会添加与 WORK_ITEM 视图的连接。此类查询将返回所有对象的所有数据。

query 和 queryAll 方法的示例

这些示例说明处理查询时生成的各种典型 API 查询及相关 SQL 语句的语法。

示例：查询处于就绪状态的任务：

本示例说明如何使用 query 方法来检索登录用户能够处理的任务。

John Smith 希望获取已分配给他的任务的清单。要使用户能够处理某个任务，该任务必须处于就绪状态。登录用户还必须拥有该任务的潜在所有者工作项。以下代码段说明此查询的 query 方法调用：

```
query( "DISTINCT TASK.TKIID",
      "TASK.KIND IN ( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING )
      AND " +
```

```
"TASK.STATE = TASK.STATE.STATE_READY AND " +
"WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
(String)null, (String)null, (Integer)null, (TimeZone)null )
```

生成 SQL SELECT 语句时，将执行下列操作：

- 对 WHERE 子句添加访问控制条件。此示例假定未启用组工作项。
- 将常量（例如 TASK.STATE.STATE_READY）替换为它们的数字值。
- 添加 FROM 子句和连接条件。

以下代码段说明根据 API 查询生成的 SQL 语句：

```
SELECT DISTINCT TASK.TKIID
FROM   TASK TA, WORK_ITEM WI,
WHERE  WI.OBJECT_ID = TA.TKIID
AND    TA.KIND IN ( 101, 105 )
AND    TA.STATE = 2
AND    WI.REASON = 1
AND    ( WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID = null AND WI.EVERYBODY = true )
```

要对 API 查询进行限制，以便只查询特定流程（例如 sampleProcess）的任务，查询如下所示：

```
query( "DISTINCT TASK.TKIID",
      "PROCESS_TEMPLATE.NAME = 'sampleProcess' AND "+
      "TASK.KIND IN ( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING )
      AND " +
      "TASK.STATE = TASK.STATE.STATE_READY AND " +
      "WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
      (String)null, (String)null, (Integer)null, (TimeZone)null )
```

示例：查询处于“已声明”状态的任务：

本示例说明如何使用 query 方法来检索登录用户所声明的任务。

用户 John Smith 想要搜索他所声明并仍处于“已声明”状态的任务。指定“由 John Smith 声明”的条件是 TASK.OWNER = 'JohnSmith'。以下代码段说明此查询的 query 方法调用：

```
query( "DISTINCT TASK.TKIID",
      "TASK.STATE = TASK.STATE.STATE_CLAIMED AND " +
      "TASK.OWNER = 'JohnSmith'",
      (String)null, (String)null, (Integer)null, (TimeZone)null )
```

以下代码段说明根据 API 查询生成的 SQL 语句：

```
SELECT DISTINCT TASK.TKIID
FROM   TASK TA, WORK_ITEM WI,
WHERE  WI.OBJECT_ID = TA.TKIID
AND    TA.STATE = 8
AND    TA.OWNER = 'JohnSmith'
AND    ( WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID = null AND WI.EVERYBODY = true )
```

声明任务时，将为该任务的所有者创建工作项。因此，查询 John Smith 所声明的任务的另一种方法是，在查询中添加以下条件以代替使用 TASK.OWNER = 'JohnSmith'：

```
WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_OWNER
```

于是，此查询将类似于以下代码段：

```
query( "DISTINCT TASK.TKIID",
      "TASK.STATE = TASK.STATE.STATE_CLAIMED AND " +
      "WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_OWNER",
      (String)null, (String)null, (Integer)null, (TimeZone)null )
```

生成 SQL SELECT 语句时，将执行下列操作：

- 对 WHERE 子句添加访问控制条件。此示例假定未启用组工作项。
- 将常量（例如 TASK.STATE.STATE_READY）替换为它们的数字值。
- 添加 FROM 子句和连接条件。

以下代码段说明根据 API 查询生成的 SQL 语句：

```
SELECT DISTINCT TASK.TKIID
FROM   TASK TA, WORK_ITEM WI,
WHERE  WI.OBJECT_ID = TA.TKIID
AND    TA.STATE = 8
AND    WI.REASON = 4
AND    ( WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID = null AND WI.EVERYBODY = true )
```

John 将要休假，因此团队领导 Anne Grant 想要检查他的当前工作量。Anne 具有系统管理员权限。她调用的查询与 John 调用的查询相同。但是，由于 Anne 是管理员，因此生成的 SQL 语句不同。以下代码段显示了所生成的 SQL 语句：

```
SELECT DISTINCT TASK.TKIID
FROM   TASK TA, WORK_ITEM WI,
WHERE  TA.TKIID = WI.OBJECT_ID =
AND    TA.STATE = 8
AND    TA.OWNER = 'JohnSmith')
```

由于 Anne 是管理员，因此未对 WHERE 子句添加访问控制条件。

示例：查询升级：

本示例说明如何使用 query 方法来检索登录用户的升级。

将任务升级时，将创建升级接收者工作项。用户 Mary Jones 想要查看已升级给她的任务的清单。以下代码段说明此查询的 query 方法调用：

```
query( "DISTINCT ESCALATION.ESIID, ESCALATION.TKIID",
      "WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_ESCALATION_RECEIVER",
      (String)null, (String)null, (Integer)null, (TimeZone)null )
```

生成 SQL SELECT 语句时，将执行下列操作：

- 对 WHERE 子句添加访问控制条件。此示例假定未启用组工作项。
- 将常量（例如 TASK.STATE.STATE_READY）替换为它们的数字值。
- 添加 FROM 子句和连接条件。

以下代码段说明根据 API 查询生成的 SQL 语句：

```
SELECT DISTINCT ESCALATION.ESIID, ESCALATION.TKIID
FROM   ESCALATION ESC, WORK_ITEM WI
WHERE  ESC.ESIID = WI.OBJECT_ID
AND    WI.REASON = 10
AND    ( WI.OWNER_ID = 'MaryJones' OR WI.OWNER_ID = null AND WI.EVERYBODY = true )
```

示例：使用 queryAll 方法：

本示例说明如何使用 queryAll 方法来检索所有属于流程模板的活动。

queryAll 方法仅可供拥有系统管理员或系统监视员权限的用户使用。以下代码段说明此查询的 queryAll 方法调用，此调用将检索所有属于流程模板 sampleProcess 的活动：

```
queryAll( "DISTINCT ACTIVITY.AIID",
         "PROCESS_TEMPLATE.NAME = 'sampleProcess'",
         (String)null, (String)null, (Integer)null, (TimeZone)null )
```

以下代码段说明根据 API 查询生成的 SQL 查询:

```
SELECT DISTINCT ACTIVITY.AIID
FROM   ACTIVITY AI, PROCESS_TEMPLATE PT
WHERE  AI.PTID = PT.PTID
AND    PT.NAME = 'sampleProcess'
```

由于此调用由管理员执行，因此未对所生成的 SQL 语句添加访问控制条件。并且，未添加与 **WORK_ITEM** 视图的连接。这意味着此查询将检索该流程模板的所有活动，包括那些没有工作项的活动。

示例: 在查询中包括查询属性:

本示例说明如何使用 `query` 方法来检索属于业务流程的任务。此流程具有您所要搜索的查询属性。

例如，您想要搜索所有处于就绪状态并属于某个业务流程的人员任务。此流程具有值为 `CID_12345` 的查询属性 **customerID** 和名称空间。以下代码段说明此查询的 `query` 方法调用:

```
query ( " DISTINCT TASK.TKIID, TASK_TEMPL.NAME, TASK.STATE,
        PROCESS_INSTANCE.NAME",
        " QUERY_PROPERTY.NAME = 'customerID' AND " +
        " QUERY_PROPERTY.STRING_VALUE = 'CID_12345' AND " +
        " QUERY_PROPERTY.NAMESPACE =
        'http://www.ibm.com/xmlns/prod/websphere/mqwf/bpel/' AND " +
        " TASK.KIND IN
        ( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING ) AND " +
        " TASK.STATE = TASK.STATE.STATE_READY ",
        (String)null, (String)null, (Integer)null, (TimeZone)null );
```

现在，如果要对查询添加第二个查询属性（例如 **Priority**）以及给定的名称空间，那么此查询的 `query` 方法调用将如下所示:

```
query ( " DISTINCT TASK.TKIID, TASK_TEMPL.NAME, TASK.STATE,
        PROCESS_INSTANCE.NAME",
        " QUERY_PROPERTY1.NAME = 'customerID' AND " +
        " QUERY_PROPERTY1.STRING_VALUE = 'CID_12345' AND " +
        " QUERY_PROPERTY1.NAMESPACE =
        'http://www.ibm.com/xmlns/prod/websphere/mqwf/bpel/' AND " +
        " QUERY_PROPERTY2.NAME = 'Priority' AND " +
        " QUERY_PROPERTY2.NAMESPACE =
        'http://www.ibm.com/xmlns/prod/websphere/mqwf/bpel/' AND " +
        " TASK.KIND IN
        ( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING ) AND " +
        " TASK.STATE = TASK.STATE.STATE_READY ",
        (String)null, (String)null, (Integer)null, (TimeZone)null );
```

如果对查询添加多个查询属性，那么必须对所添加的每个属性进行编号，如以上代码段所示。但是，查询定制属性会影响性能；性能将随所查询定制属性数的增加而下降。

示例: 在查询中包括定制属性:

本示例说明如何使用 `query` 方法来检索具有定制属性的任务。

例如，您想要搜索所有处于就绪状态并且定制属性 **customerID** 值为 CID_12345 的人员任务。以下代码段说明此查询的 `query` 方法调用：

```
query ( " DISTINCT TASK.TKIID ",
        " TASK_CPROP.NAME = 'customerID' AND " +
        " TASK_CPROP.STRING_VALUE = 'CID_12345' AND " +
        " TASK.KIND IN
        ( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING ) AND " +
        " TASK.STATE = TASK.STATE.STATE_READY ",
        (String)null, (String)null, (Integer)null, (TimeZone)null );
```

现在，如果您想要检索这些任务及其定制属性，那么此查询的 `query` 方法调用如下所示：

```
query ( " DISTINCT TASK.TKIID, TASK_CPROP.NAME, TASK_CPROP.STRING_VALUE",
        " TASK.KIND IN
        ( TASK.KIND.KIND_HUMAN, TASK.KIND.KIND_PARTICIPATING ) AND " +
        " TASK.STATE = TASK.STATE.STATE_READY ",
        (String)null, (String)null, (Integer)null, (TimeZone)null );
```

以下代码段说明根据此 API 查询生成的 SQL 语句：

```
SELECT DISTINCT TA.TKIID , TACP.NAME , TACP.STRING_VALUE
FROM TASK TA LEFT JOIN TASK_CPROP TACP ON (TA.TKIID = TACP.TKIID),
WORK_ITEM WI
WHERE WI.OBJECT_ID = TA.TKIID
AND TA.KIND IN ( 101, 105 )
AND TA.STATE = 2
AND (WI.OWNER_ID = 'JohnSmith' OR WI.OWNER_ID IS NULL AND WI.EVERYBODY = 1 )
```

此 SQL 语句包含 TASK 视图与 TASK_CPROP 视图之间的外连接。这意味着将检索满足 WHERE 子句的任务，即使它们不具有任何定制属性亦如此。

对业务流程对象和人员任务对象执行的查询的预定义视图

系统为业务流程对象和人员任务对象提供了预定义的数据库视图。在查询这些对象的引用数据时，可以使用这些视图。

使用预定义视图时，不需要对视图列明确添加 JOIN 谓词，系统将自动添加这些构造。您可以使用服务 API 的通用查询函数 `BusinessFlowManagerService` 或 `HumanTaskManagerService` 来查询此数据。也可以使用 `HumanTaskManagerDelegate` API 的相应方法或者由您的 `ExecutableQuery` 接口实现提供的预定义查询。

注：视图可能包含未描述的列。这些列仅限内部使用。

ACTIVITY 视图：

这个预定义数据库视图用于查询活动。

表 17. ACTIVITY 视图中的列

列名	类型	注释
PIID	标识	流程实例标识。
AIID	标识	活动实例标识。
PTID	标识	流程模板标识。
ATID	标识	活动模板标识。

表 17. ACTIVITY 视图中的列 (续)

列名	类型	注释
KIND	整型	活动的类型。可能的值是: KIND_INVOKE (21) KIND_RECEIVE (23) KIND_REPLY (24) KIND_THROW (25) KIND_RETHROW (46) KIND_TERMINATE (26) KIND_WAIT (27) KIND_COMPENSATE (29) KIND_SEQUENCE (30) KIND_EMPTY (3) KIND_SWITCH (32) KIND_WHILE (34) KIND_PICK (36) KIND_FLOW (38) KIND_SCOPE (40) KIND_SCRIPT (42) KIND_STAFF (43) KIND_ASSIGN (44) KIND_CUSTOM (45) KIND_FOR_EACH_PARALLEL (49) KIND_FOR_EACH_SERIAL (47)
COMPLETED	时间戳记	活动的完成时间。
ACTIVATED	时间戳记	活动的激活时间。
FIRST_ACTIVATED	时间戳记	活动的首次激活时间。
STARTED	时间戳记	活动的启动时间。
STATE	整型	活动的状态。可能的值是: STATE_INACTIVE (1) STATE_READY (2) STATE_RUNNING (3) STATE_PROCESSING_UNDO (14) STATE_SKIPPED (4) STATE_FINISHED (5) STATE_FAILED (6) STATE_TERMINATED (7) STATE_CLAIMED (8) STATE_TERMINATING (9) STATE_FAILING (10) STATE_WAITING (11) STATE_EXPIRED (12) STATE_STOPPED (13)
OWNER	字符串	所有者的主体标识。
DESCRIPTION	字符串	如果活动模板描述包含占位符, 那么此列包含解析占位符后的活动实例描述。
TEMPLATE_NAME	字符串	相关活动模板的名称。

表 17. *ACTIVITY* 视图中的列 (续)

列名	类型	注释
TEMPLATE_DESCR	字符串	相关活动模板的描述。
BUSINESS_RELEVANCE	布尔值	指定活动是否与业务相关。可能的值是： TRUE 此活动与业务相关。您可以在业务流程编排器资源管理器中查看此活动的状态。 FALSE 此活动与业务无关。
EXPIRES	时间戳记	此活动的到期日期和时间。如果此活动已到期，那么此列包含此事件的发生日期和时间。

ACTIVITY_ATTRIBUTE 视图:

这个预定义数据库视图用于查询活动的定制属性。

表 18. *ACTIVITY_ATTRIBUTE* 视图中的列

列名	类型	注释
AIID	标识	定制属性所属活动实例的标识。
NAME	字符串	定制属性的名称。
VALUE	字符串	定制属性的值。

ACTIVITY_SERVICE 视图:

这个预定义数据库视图用于查询活动服务。

表 19. *ACTIVITY_SERVICE* 视图中的列

列名	类型	注释
EIID	标识	事件实例的标识。
AIID	标识	正在等待事件的活动实例的标识。
PIID	标识	事件所属流程实例的标识。
VTID	标识	描述了事件的服务模板的标识。
PORT_TYPE	字符串	端口类型的名称。
NAME_SPACE_URI	字符串	名称空间的 URI。
OPERATION	字符串	服务的操作名。

APPLICATION_COMP 视图:

这个预定义数据库视图用于查询任务的应用程序组件标识和缺省设置。

表 20. *APPLICATION_COMP* 视图中的列

列名	类型	注释
ACOID	字符串	应用程序组件的标识。

表 20. APPLICATION_COMP 视图中的列 (续)

列名	类型	注释
BUSINESS_ RELEVANCE	布尔值	与任务业务相关的组件缺省策略。任务模板或任务中的定义可以覆盖此值。此属性影响对审计跟踪进行的日志记录。可能的值是: TRUE 任务与业务相关并且进行审计。 FALSE 任务与业务无关并且不进行审计。
NAME	字符串	应用程序组件的名称。
SUPPORT_ AUTOCLAIM	布尔值	组件的缺省自动声明策略。如果此属性设置为 TRUE , 那么当单个用户是潜在所有者时, 可以自动声明该任务。任务模板或任务中的定义可以覆盖此值。
SUPPORT_CLAIM_ SUSP	布尔值	用于确定是否可以声明已暂挂任务的组件缺省设置。如果此属性设置为 TRUE , 那么可以声明已暂挂的任务。任务模板或任务中的定义可以覆盖此值。
SUPPORT_ DELEGATION	布尔值	组件的缺省任务授权策略。如果此属性设置为 TRUE , 那么可以修改对任务指定的工作项。这表示可以创建、删除或移交工作项。
SUPPORT_FOLLOW_ON	布尔值	组件的缺省后续任务策略。如果此属性设置为 TRUE , 那么可以创建任务的后续任务。任务模板或任务中的定义可以覆盖此值。
SUPPORT_ SUB_TASK	布尔值	组件的缺省子任务策略。如果此属性设置为 TRUE , 那么可以创建任务的子任务。任务模板或任务中的定义可以覆盖此值。

ESCALATION 视图:

这个预定义数据库视图用于查询升级数据。

表 21. ESCALATION 视图中的列

列名	类型	注释
ESIID	字符串	升级实例的标识。
ACTION	整型	升级触发的操作。可能的值是: ACTION_CREATE_WORK_ITEM (1) 为每个升级接收者创建工作项。 ACTION_SEND_EMAIL (2) 向每个升级接收者发送电子邮件。 ACTION_CREATE_EVENT (3) 创建并发布事件。

表 21. ESCALATION 视图中的列 (续)

列名	类型	注释
ACTIVATION_STATE	整型	<p>如果相应任务进入下列其中一种状态，那么创建升级实例：</p> <p>ACTIVATION_STATE_READY (2) 指定人员任务或参与任务已经为声明作好准备。</p> <p>ACTIVATION_STATE_RUNNING (3) 指定发端任务已启动并且正在运行。</p> <p>ACTIVATION_STATE_CLAIMED (8) 指定已声明任务。</p> <p>ACTIVATION_STATE_WAITING_FOR_SUBTASK (20) 指定此任务正在等待子任务完成。</p>
ACTIVATION_TIME	时间戳记	升级的激活时间。
AT_LEAST_EXP_STATE	整型	<p>升级所需的任务状态。如果发生超时，那么会将任务状态与此属性的值进行比较。可能的值是：</p> <p>AT_LEAST_EXPECTED_STATE_CLAIMED (8) 指定已声明任务。</p> <p>AT_LEAST_EXPECTED_STATE_ENDED (20) 指定任务处于最终状态（已完成、已失败、已终止或已到期）。</p> <p>AT_LEAST_EXPECTED_STATE_SUBTASKS_COMPLETED (21) 指定此任务的所有子任务均已完成。</p>
ESTID	字符串	相应升级模板的标识。
FIRST_ESIID	字符串	升级链中第一个升级的标识。
INCREASE_PRIORITY	整型	<p>指示如何提高任务优先级。可能的值是：</p> <p>INCREASE_PRIORITY_NO (1) 不提高任务优先级。</p> <p>INCREASE_PRIORITY_ONCE (2) 每次将任务优先级提高一级。</p> <p>INCREASE_PRIORITY_REPEATED (3) 每当升级重复时，将任务优先级提高一级。</p>
NAME	字符串	升级的名称。
STATE	整型	<p>升级的状态。可能的值是：</p> <p>STATE_INACTIVE (1) STATE_WAITING (2) STATE_ESCALATED (3) STATE_SUPERFLUOUS (4)</p>
TKIID	字符串	升级所属任务实例的标识。

ESCALATION_CPROP 视图：

这个预定义数据库视图用于查询升级的定制属性。

表 22. *ESCALATION_CPROP* 视图中的列

列名	类型	注释
ESIID	字符串	升级标识。
NAME	字符串	属性的名称。
DATA_TYPE	字符串	非字符串定制属性类的类型。
STRING_VALUE	字符串	字符串类型的定制属性值。

ESCALATION_DESC 视图:

这个预定义数据库视图用于查询升级的多语言描述性数据。

表 23. *ESCALATION_DESC* 视图中的列

列名	类型	注释
ESIID	字符串	升级标识。
LOCALE	字符串	与描述或显示名相关联的语言环境名。
DESCRIPTION	字符串	任务模板的描述。
DISPLAY_NAME	字符串	升级的描述性名称。

ESC_TEMPL 视图:

这个预定义数据库视图用于查询升级模板的数据。

表 24. *ESC_TEMPL* 视图中的列

列名	类型	注释
ESTID	字符串	升级模板的标识。
ACTION	整型	升级触发的操作。可能的值是: ACTION_CREATE_WORK_ITEM (1) 为每个升级接收者创建工作项。 ACTION_SEND_EMAIL (2) 向每个升级接收者发送电子邮件。 ACTION_CREATE_EVENT (3) 创建并发布事件。
ACTIVATION_STATE	整型	如果相应任务进入下列其中一种状态, 那么创建升级实例: ACTIVATION_STATE_READY (2) 指定人员任务或参与任务已经为声明作好准备。 ACTIVATION_STATE_RUNNING (3) 指定发端任务已启动并且正在运行。 ACTIVATION_STATE_CLAIMED (8) 指定已声明任务。 ACTIVATION_STATE_WAITING_FOR_SUBTASK (20) 指定此任务正在等待子任务完成。

表 24. ESC_TEMPL 视图中的列 (续)

列名	类型	注释
AT_LEAST_EXP_STATE	整型	升级所需的任务状态。如果发生超时，那么会将任务状态与此属性的值进行比较。可能的值是： AT_LEAST_EXPECTED_STATE_CLAIMED (8) 指定已声明任务。 AT_LEAST_EXPECTED_STATE_ENDED (20) 指定任务处于最终状态（已完成、已失败、已终止或已到期）。 AT_LEAST_EXPECTED_STATE_SUBTASKS_COMPLETED (21) 指定此任务的所有子任务均已完成。
CONTAINMENT_CTX_ID	字符串	如果此升级模板属于直接插入任务模板，那么外层上下文是流程模板。如果升级模板上下文属于独立任务模板，那么外层上下文是任务模板。
FIRST_ESTID	字符串	升级模板链中第一个升级模板的标识。
INCREASE_PRIORITY	整型	指示如何提高任务优先级。可能的值是： INCREASE_PRIORITY_NO (1) 不提高任务优先级。 INCREASE_PRIORITY_ONCE (2) 每次将任务优先级提高一级。 INCREASE_PRIORITY_REPEATED (3) 每当升级重复时，将任务优先级提高一级。
NAME	字符串	升级模板的名称。
PREVIOUS_ESTID	字符串	升级模板链中上一个升级模板的标识。
TKTID	字符串	升级模板所属任务模板的标识。

ESC_TEMPL_CPROP 视图:

这个预定义数据库视图用于查询升级模板的定制属性。

表 25. ESC_TEMPL_CPROP 视图中的列

列名	类型	注释
ESTID	字符串	升级模板的标识。
NAME	字符串	属性的名称。
TKTID	字符串	升级模板所属任务模板的标识。
DATA_TYPE	字符串	非字符串定制属性类的类型。
VALUE	字符串	字符串类型的定制属性值。

ESC_TEMPL_DESC 视图:

这个预定义数据库视图用于查询升级模板的多语言描述性数据。

表 26. ESC_TEMPL_DESC 视图中的列

列名	类型	注释
ESTID	字符串	升级模板的标识。
LOCALE	字符串	与描述或显示名相关联的语言环境名。
TKTID	字符串	升级模板所属任务模板的标识。
DESCRIPTION	字符串	任务模板的描述。
DISPLAY_NAME	字符串	升级的描述性名称。

PROCESS_ATTRIBUTE 视图:

这个预定义数据库视图用于查询流程的定制属性。

表 27. PROCESS_ATTRIBUTE 视图中的列

列名	类型	注释
PIID	标识	定制属性所属流程实例的标识。
NAME	字符串	定制属性的名称。
VALUE	字符串	定制属性的值。

PROCESS_INSTANCE 视图:

这个预定义数据库视图用于查询流程实例。

表 28. PROCESS_INSTANCE 视图中的列

列名	类型	注释
PTID	标识	流程模板标识。
PIID	标识	流程实例标识。
NAME	字符串	流程实例的名称。
STATE	整型	流程实例的状态。可能的值是: STATE_READY (1) STATE_RUNNING (2) STATE_FINISHED (3) STATE_COMPENSATING (4) STATE_INDOUBT (10) STATE_FAILED (5) STATE_TERMINATED (6) STATE_COMPENSATED (7) STATE_COMPENSATION_FAILED (12) STATE_TERMINATING (8) STATE_FAILING (9) STATE_SUSPENDED (11)
CREATED	时间戳记	流程实例的创建时间。
STARTED	时间戳记	流程实例的启动时间。
COMPLETED	时间戳记	流程实例的完成时间。
PARENT_PIID	标识	父流程实例的标识。
PARENT_NAME	字符串	父流程实例的名称。

表 28. *PROCESS_INSTANCE* 视图中的列 (续)

列名	类型	注释
TOP_LEVEL_PIID	标识	顶层流程实例的流程实例标识。如果没有顶层流程实例，那么这是当前流程实例的流程实例标识。
TOP_LEVEL_NAME	字符串	顶层流程实例的名称。如果没有顶层流程实例，这将是当前流程实例的名称。
STARTER	字符串	流程实例的启动者的主体标识。
DESCRIPTION	字符串	如果流程模板描述包含占位符，那么此列包含解析占位符后的流程实例描述。
TEMPLATE_NAME	字符串	相关流程模板的名称。
TEMPLATE_DESCR	字符串	相关流程模板的描述。
RESUMES	时间戳记	流程实例的自动恢复时间。

***PROCESS_TEMPLATE* 视图:**

这个预定义数据库视图用于查询流程模板。

表 29. *PROCESS_TEMPLATE* 视图中的列

列名	类型	注释
PTID	标识	流程模板标识。
NAME	字符串	流程模板的名称。
VALID_FROM	时间戳记	流程模板能够被实例化的开始时间。
TARGET_NAMESPACE	字符串	流程模板的目标名称空间。
APPLICATION_NAME	字符串	流程模板所属企业应用程序的名称。
VERSION	字符串	用户定义的版本。
CREATED	时间戳记	在数据库中创建流程模板的时间。
STATE	整型	指定流程模板是否可用于创建流程实例。可能的值是: STATE_STARTED (1) STATE_STOPPED (2)
EXECUTION_MODE	整型	指定流程模板所派生的流程实例的运行方式。可能的值是: EXECUTION_MODE_MICROFLOW (1) EXECUTION_MODE_LONG_RUNNING (2)
DESCRIPTION	字符串	流程模板的描述。
COMP_SPHERE	整型	指定流程模板中微流程实例的补偿行为; 即, 是连接现有的补偿范围还是创建补偿范围。 可能的值是: COMP_SPHERE_REQUIRED (2) COMP_SPHERE_SUPPORTS (4)
DISPLAY_NAME	字符串	流程的描述性名称。

***QUERY_PROPERTY* 视图:**

这个预定义数据库视图用于查询流程级变量。

表 30. *QUERY_PROPERTY* 视图中的列

列名	类型	注释
PIID	标识	流程实例标识。
VARIABLE_NAME	字符串	流程级变量的名称。
NAME	字符串	查询属性的名称。
NAMESPACE	字符串	查询属性的名称空间。
GENERIC_VALUE	字符串	未映射至任何已定义类型（ <i>STRING_VALUE</i> 、 <i>NUMBER_VALUE</i> 、 <i>DECIMAL_VALUE</i> 或 <i>TIMESTAMP_VALUE</i> ）的属性类型的字符串表示。
STRING_VALUE	字符串	如果属性类型映射至字符串类型，那么这是该字符串的值。
NUMBER_VALUE	整型	如果属性类型映射至整数类型，那么这是该整数的值。
DECIMAL_VALUE	十进制	如果属性类型映射至浮点类型，那么这是该十进制数的值。
TIMESTAMP_VALUE	时间戳记	如果属性类型映射至时间戳记类型，那么这是该时间戳记的值。

TASK 视图:

这个预定义数据库视图用于查询任务对象。

表 31. *TASK* 视图中的列

列名	类型	注释
TKIID	标识	任务实例的标识。
ACTIVATED	时间戳记	任务的激活时间。
APPLIC_DEFAULTS_ID	标识	指定了任务缺省值的应用程序组件的标识。
APPLIC_NAME	字符串	任务所属企业应用程序的名称。
BUSINESS_RELEVANCE	布尔值	指定任务是否与业务相关。此属性影响对审计跟踪进行的日志记录。可能的值是： TRUE 任务与业务相关并且进行审计。 FALSE 任务与业务无关并且不进行审计。
COMPLETED	时间戳记	任务的完成时间。
CONTAINMENT_CTX_ID	标识	任务的外围上下文。此属性确定任务的生命周期。删除任务的外围上下文时，也将删除该任务。

表 31. TASK 视图中的列 (续)

列名	类型	注释
CTX_ AUTHORIZATION	整型	允许任务所有者访问任务上下文。可能的值是: AUTH_NONE 不需要相关上下文对象的权限。 AUTH_READER 要对相关上下文对象执行操作, 需要阅读者权限 (例如, 读取流程实例的属性)。
DUE	时间戳记	任务的到期时间。
EXPIRES	时间戳记	任务的到期日期。
FIRST_ACTIVATED	时间戳记	任务的第一次激活时间。
FOLLOW_ON_TKIID	标识	后续任务的实例标识。
HIERARCHY_ POSITION	整型	可能的值是: HIERARCHY_POSITION_TOP_TASK (0) 此任务是任务层次结构中的顶层任务。 HIERARCHY_POSITION_SUB_TASK (1) 此任务是任务层次结构中的子任务。 HIERARCHY_POSITION_FOLLOW_ON_TASK (2) 此任务是任务层次结构中的后续任务。
IS_AD_HOC	布尔值	指示此任务是在运行时以动态方式创建还是根据任务模板创建。
IS_ESCALATED	布尔值	指示任务的升级是否已发生。
IS_INLINE	布尔值	指示任务是否是业务流程中的直接插入任务。
IS_WAIT_FOR_SUB_TK	布尔值	指示父任务是否等待子任务进入结束状态。
KIND	整型	任务的类型。可能的值是: KIND_HUMAN (101) 指示任务是由人员创建和处理的协作任务。 KIND_WPC_STAFF_ACTIVITY (102) 指示任务是一个人员任务, 该任务是 WebSphere Business Integration Server Foundation V5 业务流程的人员活动。 KIND_ORIGINATING (103) 指示任务是支持人员到计算机的交互的调用任务, 此任务使人员能够创建和启动服务。 KIND_PARTICIPATING (105) 指示任务是支持计算机到人员的交互的待执行任务, 此任务使人员能够实现服务。 KIND_ADMINISTRATIVE (106) 指示任务是管理任务。
LAST_MODIFIED	时间戳记	任务的上次修改时间。
LAST_STATE_CHANGE	时间戳记	任务状态的上次修改时间。
NAME	字符串	任务的名称。

表 31. TASK 视图中的列 (续)

列名	类型	注释
NAME_SPACE	字符串	用于对任务进行分类的名称空间。
ORIGINATOR	字符串	任务发起者的主体标识。
OWNER	字符串	任务所有者的主体标识。
PARENT_CONTEXT_ID	字符串	任务的父上下文。此属性提供主调应用程序组件中相应上下文的键。父上下文由创建任务的应用程序组件设置。
PRIORITY	整型	任务的优先级。
RESUMES	时间戳记	任务的自动恢复时间。
STARTED	时间戳记	任务的启动时间 (STATE_RUNNING 和 STATE_CLAIMED)。
STARTER	字符串	任务启动者的主体标识。
STATE	整型	任务的状态。可能的值是: STATE_READY (2) 指示已准备好声明任务。 STATE_RUNNING (3) 指示任务已启动并正在运行。 STATE_FINISHED (5) 指示任务已成功完成。 STATE_FAILED (6) 指示任务未成功完成。 STATE_TERMINATED (7) 指示任务已由于外部或内部请求而终止。 STATE_CLAIMED (8) 指示任务已被声明。 STATE_EXPIRED (12) 指示任务已由于超出指定的持续时间而结束。 STATE_FORWARDED (101) 指示任务已随后续任务一起完成。
SUPPORT_AUTOCLAIM	布尔值	指示任务在被分配给单一用户时是否自动被声明。
SUPPORT_CLAIM_SUSP	布尔值	指示任务在暂挂后是否能够被声明。
SUPPORT_DELEGATION	布尔值	指示任务是否支持通过创建、删除或移交工作项来进行工作委托。
SUPPORT_FOLLOW_ON	布尔值	指示此任务是否支持创建后续任务。
SUPPORT_SUB_TASK	布尔值	指示此任务是否支持创建子任务。
SUSPENDED	布尔值	指示任务是否已暂挂。
TKTID	标识	任务模板标识。
TOP_TKIID	标识	对于子任务, 这是顶层父任务的实例标识。
TYPE	字符串	用于对任务进行分类的类型。

TASK_CPROP 视图:

这个预定义数据库视图用于查询任务对象的定制属性。

表 32. TASK_CPROP 视图中的列

列名	类型	注释
TKIID	字符串	任务实例标识。
NAME	字符串	属性的名称。
DATA_TYPE	字符串	非字符串定制属性类的类型。
STRING_VALUE	字符串	字符串类型的定制属性值。

TASK_DESC 视图:

这个预定义数据库视图用于查询任务对象的多语言描述性数据。

表 33. TASK_DESC 视图中的列

列名	类型	注释
TKIID	字符串	任务实例标识。
LOCALE	字符串	与描述或显示名相关联的语言环境名。
DESCRIPTION	字符串	任务的描述。
DISPLAY_NAME	字符串	任务的描述性名称。

TASK_TEMPL 视图:

这个预定义数据库视图包含用于将任务实例化的数据。

表 34. TASK_TEMPL 视图中的列

列名	类型	注释
TKTID	字符串	任务模板标识。
VALID_FROM	时间戳记	任务模板变为可用于实例化的时间。
APPLIC_DEFAULTS_ID	字符串	指定了任务模板缺省值的应用程序组件的标识。
APPLIC_NAME	字符串	任务模板所属企业应用程序的名称。
BUSINESS_RELEVANCE	布尔值	指定任务模板是否与业务相关。此属性影响对审计跟踪进行的日志记录。可能的值是： TRUE 任务与业务相关并且进行审计。 FALSE 任务与业务无关并且不进行审计。
CONTAINMENT_CTX_ID	标识	任务模板的外围上下文。此属性确定任务模板的生命周期。删除外围上下文时，也将删除该任务模板。

表 34. TASK_TEMPL 视图中的列 (续)

列名	类型	注释
CTX_ AUTHORIZATION	整型	允许任务所有者访问任务上下文。可能的值是: AUTH_NONE 不需要相关上下文对象的权限。 AUTH_READER 要对相关上下文对象执行操作, 需要阅读者权限 (例如, 读取流程实例的属性)。
DEFINITION_NAME	字符串	任务执行语言 (TEL) 文件中任务模板定义的名称。
DEFINITION_NS	字符串	TEL 文件中任务模板定义的名称空间。
IS_AD_HOC	布尔值	指示此任务模板是在运行时以动态方式创建还是在将任务作为 EAR 文件组成部分进行部署时创建。
IS_INLINE	布尔值	指示任务模板是否作为业务流程中的任务进行建模。
KIND	整型	任务模板所派生的任务的类型。可能的值是: KIND_HUMAN (101) 指示任务是由人员创建和处理的协作任务。 KIND_ORIGINATING (103) 指示任务是支持人员到计算机的交互的调用任务, 此任务使人员能够创建和启动服务。 KIND_PARTICIPATING (105) 指示任务是支持计算机到人员的交互的待执行任务, 此任务使人员能够实现服务。 KIND_ADMINISTRATIVE (106) 指示任务是管理任务。
NAME	字符串	任务模板的名称。
NAMESPACE	字符串	用于对任务模板进行分类的名称空间。
PRIORITY	整型	任务模板的优先级。
STATE	整型	任务模板的状态。可能的值是: STATE_STARTED (1) 指定任务模板可用于创建任务实例。 STATE_STOPPED (2) 指定任务模板已停止。在此状态下, 无法根据任务模板来创建任务实例。
SUPPORT_ AUTOCLAIM	布尔值	指示任务模板所派生的任务在被分配给单一用户时是否能够被自动声明。
SUPPORT_ CLAIM_ SUSP	布尔值	指示任务模板所派生的任务在暂挂后是否能够被声明。
SUPPORT_ DELEGATION	布尔值	指示任务模板所派生的任务是否支持通过创建、删除或移交工作项进行工作委托。
SUPPORT_ FOLLOW_ON	布尔值	指示此任务模板是否支持创建后续任务。
SUPPORT_ SUB_TASK	布尔值	指示此任务模板是否支持创建子任务。
TYPE	字符串	用于对任务模板进行分类的类型。

TASK_TEMPL_CPROP 视图:

这个预定义数据库视图用于查询任务模板的定制属性。

表 35. TASK_TEMPL_CPROP 视图中的列

列名	类型	注释
TKTID	字符串	任务模板标识。
NAME	字符串	属性的名称。
DATA_TYPE	字符串	非字符串定制属性类的类型。
STRING_VALUE	字符串	字符串类型的定制属性值。

TASK_TEMPL_DESC 视图:

这个预定义数据库视图用于查询任务模板对象的多语言描述性数据。

表 36. TASK_TEMPL_DESC 视图中的列

列名	类型	注释
TKTID	字符串	任务模板标识。
LOCALE	字符串	与描述或显示名相关联的语言环境名。
DESCRIPTION	字符串	任务模板的描述。
DISPLAY_NAME	字符串	任务模板的描述性名称。

WORK_ITEM 视图:

这个预定义数据库视图用于查询流程、任务和升级的工作项和授权数据。

表 37. WORK_ITEM 视图中的列

列名	类型	注释
WIID	标识	工作项标识。
OWNER_ID	字符串	所有者的主体标识。
GROUP_NAME	字符串	相关联的组工作列表的名称。
EVERYBODY	布尔值	指定是否每个人都拥有此工作项。

表 37. WORK_ITEM 视图中的列 (续)

列名	类型	注释
OBJECT_TYPE	整型	<p>相关联的对象的类型。可能的值是:</p> <p>OBJECT_TYPE_ACTIVITY (1) 指定此工作项是为活动创建的。</p> <p>OBJECT_TYPE_PROCESS_INSTANCE (3) 指定此工作项是为流程实例创建的。</p> <p>OBJECT_TYPE_TASK_INSTANCE (5) 指定此工作项是为任务创建的。</p> <p>OBJECT_TYPE_TASK_TEMPLATE (6) 指定此工作项是为任务模板创建的。</p> <p>OBJECT_TYPE_ESCALATION_INSTANCE (7) 指定此工作项是为升级实例创建的。</p> <p>OBJECT_TYPE_APPLICATION_COMPONENT (9) 指定此工作项是为应用程序组件创建的。</p>
OBJECT_ID	标识	相关联的对象（例如相关联的流程或任务）的标识。
ASSOC_OBJECT_TYPE	整型	ASSOC_OID 属性所引用的对象的类型，例如任务、流程或外部对象。使用 OBJECT_TYPE 属性的值。
ASSOC_OID	标识	与工作项相关联的对象的标识。例如，此工作项所属活动实例的相关流程实例标识（PIID）。
REASON	整型	<p>此工作项的分配原因。可能的值是:</p> <p>REASON_POTENTIAL_STARTER (5)</p> <p>REASON_POTENTIAL_INSTANCE_CREATOR (11)</p> <p>REASON_POTENTIAL_STARTER (1)</p> <p>REASON_EDITOR (2)</p> <p>REASON_READER (3)</p> <p>REASON_ORIGINATOR (9)</p> <p>REASON_OWNER (4)</p> <p>REASON_STARTER (6)</p> <p>REASON_ESCALATION_RECEIVER (10)</p> <p>REASON_ADMINISTRATOR (7)</p>
CREATION_TIME	时间戳记	工作项的创建日期和时间。

在查询中使用变量对数据进行过滤

查询结果返回与查询条件匹配的对象。您可能想根据变量值对这些结果进行过滤。

关于此任务

您可以定义变量，以供流程模型中的流程在运行时使用。对于这些变量，必须声明可以查询的部件。

例如, John Smith 拨打了保险公司的热线电话, 以了解他的车辆受损保险索赔进度。索赔管理员使用客户标识来查找该项索赔。

过程

1. 可选: 列示流程中可查询变量的属性。

使用流程模板标识来标识流程。如果您知道可查询变量, 那么可以跳过此步骤。

```
List variableProperties = process.getQueryProperties(ptid);
for (int i = 0; i < variableProperties.size(); i++)
{
    QueryProperty queryData = (QueryProperty)variableProperties.get(i);
    String variableName = queryData.getVariableName();
    String name         = queryData.getName();
    int mappedType     = queryData.getMappedType();
    ...
}
```

2. 列示变量与过滤器条件匹配的流程实例。

对于本流程而言, 已将客户标识设计成可查询变量 `customerClaim` 的组成部分。因此, 可以使用客户标识来查找索赔信息。

```
QueryResultSet result = process.query
    ("PROCESS_INSTANCE.NAME, QUERY_PROPERTY.STRING_VALUE",
    "QUERY_PROPERTY.VARIABLE_NAME = 'customerClaim' AND " +
    "QUERY_PROPERTY.NAME = 'customerID' AND " +
    "QUERY_PROPERTY.STRING_VALUE like 'Smith%'",
    (String)null, (Integer)null,
    (Integer)null, (TimeZone)null );
```

此操作将返回一个查询结果集, 该结果集包含客户标识以 `Smith` 开头的客户的流程实例名和值。

管理存储查询

可以将经常运行的查询保存下来, 使它们成为“存储查询”。存储查询可以是可供所有用户使用的查询, 即公用查询, 也可以是属于特定用户的查询, 即专用查询。

关于此任务

存储查询是指存储在数据库中并由名称标识的查询。专用存储查询可以与公用存储查询同名; 不同所有者的专用存储查询也可以同名。

您可以使用业务流程对象或任务对象的存储查询, 也可以同时使用这两类对象的存储查询。

管理公用存储查询

公用存储查询由系统管理员创建。这些查询可供所有用户使用。

关于此任务

系统管理员可以创建、查看和删除公用存储查询。如果在 API 调用中未指定用户标识, 那么将假定该存储查询是公用存储查询。

过程

1. 创建公用存储查询。

例如，以下代码段创建流程实例的存储查询、保存该查询并对其指定名称 CustomerOrdersStartingWithA。

```
process.createStoredQuery("CustomerOrdersStartingWithA",
    "DISTINCT PROCESS_INSTANCE.PIID, PROCESS_INSTANCE.NAME",
    "PROCESS_INSTANCE.NAME LIKE 'A%'",
    "PROCESS_INSTANCE.NAME",
    (Integer)null, (TimeZone)null);
```

此存储查询的结果是所有以字母 A 开头的流程实例名及其相关流程实例标识 (PIID) 的有序列表。

2. 运行由存储查询定义的查询。

```
QueryResultSet result = process.query("CustomerOrdersStartingWithA",
    new Integer(0));
```

此操作返回符合条件的对象。在本例中，此操作返回所有以 A 开头的客户订单。

3. 列示可用的公用存储查询的名称。

以下代码段说明如何对返回的查询列表进行限制，以便仅返回公用查询。

```
String[] storedQuery = process.getStoredQueryNames(StoredQueryData.KIND_PUBLIC);
```

4. 可选：检查由特定存储查询定义的查询。

专用存储查询可以与公用存储查询同名。如果这些名称相同，那么将返回专用存储查询。以下代码段说明如何只返回具有指定名称的公用查询。如果要对基于任务的对象运行此查询，请指定 StoredQuery 作为返回的对象类型，而不要指定 StoredQueryData。

```
StoredQueryData storedQuery = process.getStoredQuery
    (StoredQueryData.KIND_PUBLIC, "CustomerOrdersStartingWithA");
String selectClause = storedQuery.getSelectClause();
String whereClause = storedQuery.getWhereClause();
String orderByClause = storedQuery.getOrderByClause();
Integer threshold = storedQuery.getThreshold();
String owner = storedQuery.getOwner();
```

5. 删除公用存储查询。

以下代码段说明如何删除步骤 1 中创建的存储查询。

```
process.deleteStoredQuery("CustomerOrdersStartingWithA");
```

管理其他用户的专用存储查询

专用查询可以由任何用户创建。这些查询只能供该查询的所有者和系统管理员使用。

关于此任务

系统管理员可以管理属于特定用户的专用存储查询。

过程

1. 为用户标识 Smith 创建专用存储查询。

例如，以下代码段为用户标识 Smith 创建流程实例的存储查询、保存该查询并对其指定特定的名称。

```
process.createStoredQuery("Smith", "CustomerOrdersStartingWithA",
    "DISTINCT PROCESS_INSTANCE.PIID, PROCESS_INSTANCE.NAME",
    "PROCESS_INSTANCE.NAME LIKE 'A%'",
    "PROCESS_INSTANCE.NAME",
    (Integer)null, (TimeZone)null,
    (List)null, (String)null);
```

此存储查询的结果是所有以字母 A 开头的流程实例名及其相关流程实例标识 (PIID) 的有序列表。

2. 运行由存储查询定义的查询。

```
QueryResultSet result = process.query
    ("Smith", "CustomerOrdersStartingWithA",
    (Integer)null, (Integer)null, (List)null);
new Integer(0));
```

此操作返回符合条件的对象。在本例中，此操作返回所有以 A 开头的客户订单。

3. 获取属于特定用户的专用查询的名称列表。

例如，以下代码段说明如何获取属于用户 Smith 的专用查询列表。

```
String[] storedQuery = process.getStoredQueryNames("Smith");
```

4. 查看特定查询的详细信息。

以下代码段说明如何查看用户 Smith 拥有的查询 CustomerOrdersStartingWithA 的详细信息。

```
StoredQuery storedQuery = process.getStoredQuery
    ("Smith", "CustomerOrdersStartingWithA");
String selectClause = storedQuery.getSelectClause();
String whereClause = storedQuery.getWhereClause();
String orderByClause = storedQuery.getOrderByClause();
Integer threshold = storedQuery.getThreshold();
String owner = storedQuery.getOwner();
```

5. 删除专用存储查询。

以下代码段说明如何删除用户 Smith 拥有的专用查询。

```
process.deleteStoredQuery("Smith", "CustomerOrdersStartingWithA");
```

使用专用存储查询

如果您不是系统管理员，那么可以创建、运行和删除自己的专用存储查询。您还可以使用系统管理员创建的公用存储查询。

过程

1. 创建专用存储查询。

例如，以下代码段创建流程实例的存储查询、保存该查询并对其指定特定的名称。如果未指定用户标识，那么将假定该存储查询是登录用户的专用存储查询。

```
process.createStoredQuery("CustomerOrdersStartingWithA",
    "DISTINCT PROCESS_INSTANCE.PIID, PROCESS_INSTANCE.NAME",
    "PROCESS_INSTANCE.NAME LIKE 'A%'",
    "PROCESS_INSTANCE.NAME",
    (Integer)null, (TimeZone)null);
```

此查询返回所有以字母 A 开头的流程实例名及其相关流程实例标识 (PIID) 的有序列表。

2. 运行由存储查询定义的查询。

```
QueryResultSet result = process.query("CustomerOrdersStartingWithA",
    new Integer(0));
```

此操作返回符合条件的对象。在本例中，此操作返回所有以 A 开头的客户订单。

3. 获取登录用户能够访问的存储查询的名称列表。

以下代码段说明如何获取用户能够访问的公用存储查询和专用存储查询。

```
String[] storedQuery = process.getStoredQueryNames();
```

4. 查看特定查询的详细信息。

以下代码段说明如何查看用户 Smith 拥有的查询 CustomerOrdersStartingWithA 的详细信息。

```
StoredQuery storedQuery = process.getStoredQuery
    ("CustomerOrdersStartingWithA");
String selectClause = storedQuery.getSelectClause();
String whereClause = storedQuery.getWhereClause();
String orderByClause = storedQuery.getOrderByClause();
Integer threshold = storedQuery.getThreshold();
String owner = storedQuery.getOwner();
```

5. 删除专用存储查询。

以下代码段说明如何删除专用存储查询。

```
process.deleteStoredQuery("CustomerOrdersStartingWithA");
```

为业务流程开发应用程序

业务流程是一组与业务相关的活动，您按特定顺序调用这些活动以实现业务目标。下面提供的示例说明如何通过开发应用程序对流程执行典型操作。

关于此任务

业务流程可以是微流程，也可以是长时间运行流程：

- 微流程是以同步方式执行的短时间运行业务流程。在一段非常短的时间过后，会将结果返回给调用者。
- 可中断的长时间运行流程是作为一系列链接到一起的活动执行。在流程中使用某些构造会导致流程中断，例如调用人员任务、使用同步绑定调用服务或者使用计时器驱动的活动。

流程的并行分支通常以异步方式进行导航，即，并行分支中的活动以并发方式执行。根据活动的类型和事务设置的不同，该活动可以在它自己的事务中运行。

对流程实例执行的操作所需的角色

对 BusinessFlowManager 接口的访问权并不保证调用者能够对流程执行所有操作。该调用者必须使用有权执行操作的角色来登录客户机应用程序。

下表说明特定角色能够对流程实例执行的操作。

操作	调用者的主体角色		
	阅读者	启动者	管理员
createMessage	x	x	x
createWorkItem			x

操作	调用者的主体角色		
	阅读者	启动者	管理员
delete			x
deleteWorkItem			x
forceTerminate			x
getActiveEventHandlers	x		x
getActivityInstance	x		x
getAllActivities	x		x
getAllWorkItems	x		x
getClientUISettings	x	x	x
getCustomProperties	x	x	x
getCustomProperty	x	x	x
getCustomPropertyNames	x	x	x
getFaultMessage	x	x	x
getInputClientUISettings	x	x	x
getInputMessage	x	x	x
getOutputClientUISettings	x	x	x
getOutputMessage	x	x	x
getProcessInstance	x	x	x
getVariable	x	x	x
getWaitingActivities	x	x	x
getWorkItems	x		x
restart			x
resume			x
setCustomProperty		x	x
setVariable			x
suspend			x
transferWorkItem			x

对业务流程活动执行的操作所需的角色

对 BusinessFlowManager 接口的访问权并不保证调用者能够对活动执行所有操作。该调用者必须使用有权执行操作的角色来登录客户机应用程序。

下表说明特定角色能够对活动实例执行的操作。

操作	调用者的主体角色				
	阅读者	编辑者	潜在所有者	所有者	管理员
cancelClaim				x	x
claim			x		x
complete				x	x
createMessage	x	x	x	x	x
createWorkItem					x

操作	调用者的主体角色				
	阅读者	编辑者	潜在所有者	所有者	管理员
deleteWorkItem					x
forceComplete					x
forceRetry					x
getActivityInstance	x	x	x	x	x
getAllWorkItems	x	x	x	x	x
getClientUISettings	x	x	x	x	x
getCustomProperties	x	x	x	x	x
getCustomProperty	x	x	x	x	x
getCustomPropertyNames	x	x	x	x	x
getFaultMessage	x	x	x	x	x
getFaultNames	x	x	x	x	x
getInputMessage	x	x	x	x	x
getOutputMessage	x	x	x	x	x
getVariable	x	x	x	x	x
getVariableNames	x	x	x	x	x
getInputVariableNames	x	x	x	x	x
getOutputVariableNames	x	x	x	x	x
getWorkItems	x	x	x	x	x
setCustomProperty		x		x	x
setFaultMessage		x		x	x
setOutputMessage		x		x	x
setVariable					x
transferWorkItem				x 只能移交给潜在所有者或管理员	x

管理业务流程的生命周期

在调用可以启动流程的业务流程编排器 API 方法时，将创建流程实例。流程实例的导航将一直持续到它的所有活动都处于结束状态为止。您可以对流程实例执行各种操作，以便对其生命周期进行管理。

关于此任务

下面提供的示例说明如何通过开发应用程序对流程执行下列典型生命周期操作。

启动业务流程

业务流程的启动方式取决于该流程是微流程还是长时间运行流程。启动流程的服务对于流程的启动方式来说也很重要；流程可以有唯一的启动服务，也可以有多个启动服务。

关于此任务

提供的示例说明了如何为微流程和长时间运行流程的典型启动方案开发应用程序。

运行包含唯一启动服务的微流程:

微流程可以由接收或检取活动启动。如果微流程由接收活动启动，或者检取活动只有一个 `onMessage` 定义，那么启动服务是唯一的。

关于此任务

如果微流程实现了请求/响应操作，即该流程包含应答，那么可以使用 `call` 方法来运行该流程并传递流程模板名称作为调用参数。

如果微流程是单向操作，请使用 `sendMessage` 方法运行该流程。本示例未阐述此方法。

过程

1. 可选： 列示流程模板以查找所要运行的流程的名称。

如果您已知道该流程的名称，那么此步骤是可选的。

```
ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
PROCESS_TEMPLATE.EXECUTION_MODE.EXECUTION_MODE_MICROFLOW",
"PROCESS_TEMPLATE.NAME",
new Integer(50),
(TimeZone)null);
```

结果将按名称排序。此查询将返回一个数组，该数组包含 `call` 方法可以启动的前 50 个模板并已进行排序。

2. 使用适当类型的输入消息来启动流程。

创建消息时，必须指定其消息类型名，以便包含消息定义。

```
ProcessTemplateData template = processTemplates[0];
//create a message for the single starting receive activity
ClientObjectWrapper input = process.createMessage
(template.getID(),
template.getInputMessageType());
DataObject myMessage = null;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
myMessage = (DataObject)input.getObject();
//set the strings in the message, for example, a customer name
myMessage.setString("CustomerName", "Smith");
}

//run the process
ClientObjectWrapper output = process.call(template.getName(), input);
DataObject myOutput = null;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
myOutput = (DataObject)output.getObject();
int order = myOutput.getInt("OrderNo");
}
```

此操作创建流程模板 `CustomerTemplate` 的实例并传递一些客户数据。仅当该流程完成后，此操作才会返回。而且，该流程的结果 `OrderNo` 将返回给调用者。

运行包含非唯一启动服务的微流程:

微流程可以由接收或检取活动启动。如果微流程由具有多个 `onMessage` 定义的检取活动启动，那么启动服务不是唯一的。

关于此任务

如果微流程实现了请求/响应操作，即该流程包含应答，那么可以使用 `call` 方法来运行该流程并在进行调用时传递启动服务的标识。

如果微流程是单向操作，请使用 `sendMessage` 方法运行该流程。本示例未阐述此方法。

过程

1. 可选： 列示流程模板以查找所要运行的流程的名称。

如果您已知道该流程的名称，那么此步骤是可选的。

```
ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
    PROCESS_TEMPLATE.EXECUTION_MODE.EXCECUTION_MODE_MICROFLOW",
"PROCESS_TEMPLATE.NAME",
    new Integer(50),
    (TimeZone)null);
```

结果将按名称排序。此查询将返回一个数组，该数组包含前 50 个可以作为微流程启动的模板并已进行排序。

2. 确定要调用的启动服务。

本示例使用找到的第一个模板。

```
ProcessTemplateData template = processTemplates[0];
ActivityServiceTemplateData[] startActivities =
    process.getStartActivities(template.getID());
```

3. 使用适当类型的输入消息来启动流程。

创建消息时，必须指定其消息类型名，以便包含消息定义。

```
ActivityServiceTemplateData activity = startActivities[0];
//create a message for the service to be called
ClientObjectWrapper input =
    process.createMessage(activity.getServiceTemplateID(),
        activity.getActivityTemplateID(),
        activity.getInputMessageType());
DataObject myMessage = null;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
    myMessage = (DataObject)input.getObject();
    //set the strings in the message, for example, a customer name
    myMessage.setString("CustomerName", "Smith");
}
//run the process
ClientObjectWrapper output = process.call(activity.getServiceTemplateID(),
        activity.getActivityTemplateID(),
        input);
//check the output of the process, for example, an order number
DataObject myOutput = null;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
    myOutput = (DataObject)output.getObject();
    int order = myOutput.getInt("OrderNo");
}
```


此操作创建流程模板 `CustomerTemplate` 的实例并传递一些客户数据。仅当该流程完成后，此操作才会返回。而且，该流程的结果 `OrderNo` 将返回给调用者。

启动包含唯一启动服务的长时间运行流程:

如果启动服务是唯一的，那么可以使用 `initiate` 方法并传递流程模板名称作为参数。当使用单个接收或检取活动来启动长时间运行流程时，以及当单个检取活动只有一个 `onMessage` 定义时，启动服务就是唯一的。

过程

1. 可选: 列示流程模板以查找所要启动的流程的名称。

如果您已知道该流程的名称，那么此步骤是可选的。

```
ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
PROCESS_TEMPLATE.EXECUTION_MODE.EXCECUTION_MODE_LONG_RUNNING",
"PROCESS_TEMPLATE.NAME",
new Integer(50),
(TimeZone)null);
```

结果将按名称排序。此查询将返回一个数组，该数组包含 `initiate` 方法可以启动的前 50 个模板并已进行排序。

2. 使用适当类型的输入消息来启动流程。

创建消息时，必须指定其消息类型名，以便包含消息定义。如果指定流程实例名，那么该名称不能以下划线开头。如果未指定流程实例名，那么将使用字符串格式的流程实例标识 (PIID) 作为名称。

```
ProcessTemplateData template = processTemplates[0];
//create a message for the single starting receive activity
ClientObjectWrapper input = process.createMessage
(template.getID(),
template.getInputMessageType());
DataObject myMessage = null;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
myMessage = (DataObject)input.getObject();
//set the strings in the message, for example, a customer name
myMessage.setString("CustomerName", "Smith");
}
//start the process
PIID piid = process.initiate(template.getName(), "CustomerOrder", input);
```

此操作创建实例 `CustomerOrder` 并传递一些客户数据。当流程启动时，执行的操作会将新流程实例的对象标识返回给调用者。

流程实例的启动者将被设置为该请求的调用者。此人员将接收到该流程实例的工作项。在操作中将确定该流程实例的流程管理员、阅读者和编辑器，他们将接收该流程实例的工作项。将确定后续活动实例。这些后续活动实例将自动启动，或者，如果它们是人员任务、接收活动或检取活动，那么将为潜在所有者创建工作项。

启动包含非唯一启动服务的长时间运行流程:

可以通过多个始发接收或检取活动来启动长时间运行流程。可以使用 `initiate` 方法启动流程。如果启动服务不唯一（例如，流程通过多个接收或检取活动启动，或者通过具有多个 `onMessage` 定义的检取活动启动），那么必须标识要调用的服务。

过程

1. 可选： 列示流程模板以查找所要启动的流程的名称。

如果您已知道该流程的名称，那么此步骤是可选的。

```
ProcessTemplateData[] processTemplates = process.queryProcessTemplates
("PROCESS_TEMPLATE.EXECUTION_MODE =
PROCESS_TEMPLATE.EXECUTION_MODE.EXCECUTION_MODE_LONG_RUNNING",
"PROCESS_TEMPLATE.NAME",
new Integer(50),
(TimeZone)null);
```

结果将按名称排序。此查询将返回一个数组，该数组包含前 50 个可以作为长时间运行流程启动的模板并已进行排序。

2. 确定要调用的启动服务。

```
ProcessTemplateData template = processTemplates[0];
ActivityServiceTemplateData[] startActivities =
process.getStartActivities(template.getID());
```

3. 使用适当类型的输入消息来启动流程。

创建消息时，必须指定其消息类型名，以便包含消息定义。如果指定流程实例名，那么该名称不能以下划线开头。如果未指定流程实例名，那么将使用字符串格式的流程实例标识（PIID）作为名称。

```
ActivityServiceTemplateData activity = startActivities[0];
//create a message for the service to be called
ClientObjectWrapper input = process.createMessage
(activity.getServiceTemplateID(),
activity.getActivityTemplateID(),
activity.getInputMessageType());

DataObject myMessage = null;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
myMessage = (DataObject)input.getObject();
//set the strings in the message, for example, a customer name
myMessage.setString("CustomerName", "Smith");
}
//start the process
PIID piid = process.sendMessage(activity.getServiceTemplateID(),
activity.getActivityTemplateID(),
input);
```

此操作创建实例并传递一些客户数据。当流程启动时，执行的操作会将新流程实例的对象标识返回给调用者。

流程实例的启动者将被设置为该请求的调用者，它将接收该流程实例的工作项。在操作中将确定该流程实例的流程管理员、阅读者和编辑者，他们将接收该流程实例的工作项。将确定后续活动实例。这些后续活动实例将自动启动，或者，如果它们是人员任务、接收活动或检取活动，那么将为潜在所有者创建工作项。

暂挂和恢复业务流程

可以将处于“正在运行”状态的顶层长时间运行流程实例暂挂，并可以恢复该流程以使其能够完成。

开始之前

调用者必须是流程实例管理员或业务流程管理员。要暂挂流程实例，它必须处于“正在运行”状态或“即将失败”状态。

关于此任务

例如，可以先将流程实例暂挂，以便配置对该流程稍后要使用的后端系统的访问。满足该流程的先决条件后，可以恢复该流程实例。您可能还想暂挂流程以解决导致流程实例失败的问题，接着在解决问题后恢复该流程。

过程

1. 获取处于“正在运行”状态并且要暂挂的流程 CustomerOrder。

```
ProcessInstanceData processInstance =  
    process.getProcessInstance("CustomerOrder");
```

2. 暂挂该流程实例。

```
PIID piid = processInstance.getID();  
process.suspend( piid );
```

此操作将暂挂指定的顶层流程实例。该流程实例将进入“已暂挂”状态。将 `autonomy` 属性设置为 `child` 的子流程如果处于“正在运行”、“即将失败”、“正在终止”或“正在补偿”状态，那么它们也将被暂挂。还将暂挂与此流程实例相关联的直接插入任务，但不会暂挂与此流程实例相关联的独立任务。

在此状态下，已启动的活动仍可以运行到完成，例如可以完成处于“已声明”状态的人员任务活动，但不会激活新活动。

3. 恢复该流程实例。

```
process.resume( piid );
```

此操作使该流程实例及其子流程进入它们被暂挂前所处的状态。

重新启动业务流程

可以重新启动处于“已完成”、“已终止”、“已失败”或“已补偿”状态的流程实例。

开始之前

调用者必须是流程实例管理员或业务流程管理员。

关于此任务

重新启动流程实例与第一次启动流程实例类似。但是，重新启动流程实例时，流程实例标识已知，并且已有该实例的输入消息。

如果该流程有多个能够创建流程实例的接收活动或检取活动（也称为接收选项活动），那么将使用所有属于这些活动的消息来重新启动该流程实例。如果任何这些活动实现了请求/响应操作，那么将在对相关应答活动进行导航时再次发送响应。

过程

1. 获取要重新启动的流程。

```
ProcessInstanceData processInstance =  
    process.getProcessInstance("CustomerOrder");
```

2. 重新启动该流程实例。

```
PIID piid = processInstance.getID();  
process.restart( piid );
```

此操作将重新启动指定的流程实例。

终止流程实例

有时，具有流程管理员权限的用户需要终止已知处于不可恢复状态的顶层流程实例。由于流程实例将立即终止，而不会等待任何未完成的子流程或活动完成，所以，只有在异常情况下才应该终止流程实例。

过程

1. 检索要终止的流程实例。

```
ProcessInstanceData processInstance =  
    process.getProcessInstance("CustomerOrder");
```

2. 终止该流程实例。

如果要终止流程实例，可以在进行补偿的情况下终止该流程实例，也可以不进行补偿。

要在进行补偿的情况下终止该流程实例：

```
PIID piid = processInstance.getID();  
process.forceTerminate(piid, CompensationBehaviour.INVOKE_COMPENSATION);
```

要在不进行补偿的情况下终止该流程实例：

```
PIID piid = processInstance.getID();  
process.forceTerminate(piid);
```

如果在进行补偿的情况下终止流程实例，那么将象是顶层作用域发生故障那样对该流程运行补偿。如果在未进行补偿的情况下终止流程实例，那么该流程实例将立即终止，而不会等待活动、待执行任务或直接插入调用任务正常结束。

强制终止请求不会终止该流程启动的应用程序以及与该流程相关的独立任务。如果要终止这些应用程序，那么必须在流程应用程序中添加语句以明确终止该流程启动的应用程序。

删除流程实例

如果在流程模型中为流程模板设置了相应的属性，那么将从业务流程编排器数据库中自动删除已完成的流程实例。例如，您可能想将流程实例保留在数据库中，以便查询未写入审计日志的流程实例数据。但是，存储的流程实例数据不仅影响磁盘空间和性能，还会导致无法创建使用同一关联集值的流程实例。因此，应该定期从数据库中删除流程实例数据。

关于此任务

要删除流程实例，您需要具有流程管理员权限，并且该流程实例必须是顶层流程实例。

以下示例说明如何删除所有已完成的流程实例。

过程

1. 列示已完成的流程实例。

```
QueryResultSet result =  
    process.query("DISTINCT PROCESS_INSTANCE.PIID",  
                 "PROCESS_INSTANCE.STATE =  
                 PROCESS_INSTANCE.STATE.STATE_FINISHED",  
                 (String)null, (Integer)null, (TimeZone)null);
```

此操作将返回一个查询结果集，该结果集列示了已完成的流程实例。

2. 删除已完成的流程实例。

```
while (result.next() )
{
    PIID piid = (PIID) result.getOID(1);
    process.delete(piid);
}
```

此操作将从数据库中删除所选流程实例及其直接插入任务。

处理人员任务活动

业务流程中的人员任务活动通过工作项被分配给企业中的各位人员。当流程启动时，那么将为潜在所有者创建工作项。

关于此任务

激活人员任务活动时，将创建活动实例以及相关关联的待执行任务。对人员任务活动的处理以及工作项管理工作由人员任务管理器执行。活动实例的任何状态更改都将反应在任务实例中，反之亦然。

潜在所有者声明活动。此人员负责提供相关信息并完成该活动。

过程

1. 列示已登录人员能够处理的活动:

```
QueryResultSet result =
    process.query("ACTIVITY.AIID",
        "ACTIVITY.STATE = ACTIVITY.STATE.STATE_READY AND
        ACTIVITY.KIND = ACTIVITY.KIND.KIND_STAFF AND
        WORK_ITEM.REASON =
        WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
        (String)null, (Integer)null, (TimeZone)null);
```

此操作返回一个查询结果集，该结果集包含已登录人员能够处理的活动。

2. 声明要处理的活动:

```
if (result.size() > 0)
{
    result.first();
    AIID aiid = (AIID) result.getOID(1);
    ClientObjectWrapper input = process.claim(aiid);
    DataObject activityInput = null ;
    if ( input.getObject() != null && input.getObject() instanceof DataObject )
    {
        activityInput = (DataObject)input.getObject();
        // read the values
        ...
    }
}
```

声明活动时，将返回该活动的输入消息。

3. 声明处理活动后，将完成该活动。该活动可能成功完成，也可能以报告故障消息而结束。如果该活动成功，那么将传递输出消息。如果该活动不成功，那么将进入“已失败”或“已停止”状态，并且将传递故障消息。必须为这些操作创建适当的消息。创建消息时，必须指定消息类型名，以便包含消息定义。
 - a. 要成功地完成活动，请创建输出消息。

```

ActivityInstanceData activity = process.getActivityInstance(aiid);
ClientObjectWrapper output =
    process.createMessage(aiid, activity.getOutputMessageType());
DataObject myMessage = null ;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
    myMessage = (DataObject)output.getObject();
    //set the parts in your message, for example, an order number
    myMessage.setInt("OrderNo", 4711);
}

//complete the activity
process.complete(aiid, output);

```

此操作将设置包含序号的输出消息。

- b. 要在发生故障时完成活动，请创建故障消息。

```

//retrieve the faults modeled for the human task activity
List faultNames = process.getFaultNames(aiid);

//create a message of the appropriate type
ClientObjectWrapper myFault =
    process.createMessage(aiid, faultNames.get(0));

// set the parts in your fault message, for example, an error number
DataObject myMessage = null ;
if ( myFault.getObject() != null && input.getObject() instanceof DataObject )
{
    myMessage = (DataObject)myFault.getObject();
    //set the parts in the message, for example, a customer name
    myMessage.setInt("error",1304);
}

process.complete(aiid, (String)faultNames.get(0), myFault);

```

此操作设置处于“已失败”或“已停止”状态的活动。如果流程模型中该活动的 **continueOnError** 参数设置为 true，那么会将该活动置于“已失败”状态，并且导航将继续进行。如果 **continueOnError** 参数设置为 false，并且在外层作用域中未捕获该故障，那么将使该活动进入“已停止”状态。在此状态下，可以通过强制完成或强制重试来修复该活动。

处理单一人员工作流程

某些工作流程仅由一位人员执行，例如，从网上书店订购书籍。此类工作流程没有并行路径。completeAndClaimSuccessor API 支持对此类工作流程进行处理。

关于此任务

在网上书店，顾客需要完成一系列操作以订购书籍。此操作序列可以作为一系列人员任务活动（待执行任务）实现。如果顾客决定订购多本书籍，那么等同于声明下一个人员任务活动。由于用户界面定义与用于控制用户界面中对话框流的活动相关联，因此，此类工作流程也称为页面流。

completeAndClaimSuccessor API 完成一个人员任务活动并在已登录人员的流程实例中声明下一个人员任务活动。此 API 将返回关于所声明的下一个活动的信息，包括要处理的输入消息。由于下一个活动将在已完成活动所在的事务中执行，因此必须将流程模型中所有人员任务活动的事务行为设置为 participates。

请将此示例与同时使用业务流程管理器 API 和人员任务管理器 API 的示例进行比较。

过程

1. 声明活动序列中的第一个活动。

```
//
//Query the list of activities that can be claimed by the logged-on user
//
QueryResultSet result =
    process.query("ACTIVITY.AIID",
        "PROCESS_INSTANCE.NAME = 'CustomerOrder' AND
        ACTIVITY.STATE = ACTIVITY.STATE.STATE_READY AND
        ACTIVITY.KIND = ACTIVITY.KIND.KIND_STAFF AND
        WORK_ITEM.REASON =
        WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
        (String)null, (Integer)null, (TimeZone)null);
...
//
//Claim the first activity
//
if (result.size() > 0)
{
    result.first();
    AIID aaid = (AIID) result.getOID(1);
    ClientObjectWrapper input = process.claim(aaid);
    DataObject activityInput = null ;
    if ( input.getObject() != null && input.getObject() instanceof DataObject )
    {
        activityInput = (DataObject)input.getObject();
        // read the values
        ...
    }
}
```

声明活动时，将返回该活动的输入消息。

2. 该活动中的处理完成后，使该活动完成，然后声明下一个活动。

要完成活动，必须传递输出消息。创建输出消息时，必须指定消息类型名，以便包含消息定义。

```
ActivityInstanceData activity = process.getActivityInstance(aaid);
ClientObjectWrapper output =
    process.createMessage(aaid, activity.getOutputMessageType());
DataObject myMessage = null ;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
    myMessage = (DataObject)output.getObject();
    //set the parts in your message, for example, an order number
    myMessage.setInt("OrderNo", 4711);
}

//complete the activity and claim the next one
CompleteAndClaimSuccessorResult successor =
    process.completeAndClaimSuccessor(aaid, output);
```

此操作将设置包含订单号的输出消息并声明序列中的下一个活动。如果对后续活动设置了 AutoClaim，并且可以沿多条路径进行导航，那么将声明所有后续活动并随机返回一个活动作为下一个活动。如果没有其他后续活动可分配给此用户，那么将返回 Null。

如果该流程包含可以使用的并行路径，并且这些路径包含人员任务活动，已登录用户是多个此类活动的潜在所有者，那么将会自动声明随机活动并将其作为下一个活动返回。

3. 处理下一个活动。


```

String name = successor.getActivityName();

ClientObjectWrapper nextInput = successor.getInputMessage();
if ( nextInput.getObject() !=
    null && nextInput.getObject() instanceof DataObject )
{
    activityInput = (DataObject)input.getObject();
    // read the values
    ...
}

aiid = successor.getAIID();

```

4. 继续执行步骤 2 以完成该活动。

向处于“正在等待”状态的活动发送消息

可以使用入站消息活动（接收活动、检取活动中的 `onMessage` 和事件处理程序中的 `onEvent`）使运行中的流程与来自“外部世界”的事件同步。例如，接收客户发送的电子邮件以响应信息请求就是这样的事件。

关于此任务

可以使用发端任务来将消息发送至活动。

过程

1. 列示具有特定流程实例标识的流程实例中正在等待登录用户的消息的活动服务模板。

```
ActivityServiceTemplateData[] services = process.getWaitingActivities(piid);
```

2. 将消息发送至第一个处于等待状态的活动。

假定第一个服务就是要处理的服务。调用者必须是接收消息的活动的潜在启动者或者流程实例的管理员。

```

VTID vtid = services[0].getServiceTemplateID();
ATID atid = services[0].getActivityTemplateID();
String inputType = services[0].getInputMessageType();

// create a message for the service to be called
ClientObjectWrapper message =
    process.createMessage(vtid,atid,inputMessageType);
DataObject myMessage = null;
if ( message.getObject() != null && message.getObject() instanceof DataObject )
{
    myMessage = (DataObject)message.getObject();
    //set the strings in the message, for example, chocolate is to be ordered
    myMessage.setString("Order", "chocolate");
}

// send the message to the waiting activity
process.sendMessage(vtid, atid, message);
}

```

此操作将指定的消息发送至处于“正在等待”状态的活动服务并传递一些订单数据。

您还可以指定流程实例标识以确保将该消息发送至指定的流程实例。如果未指定流程实例标识，那么会将该消息发送至活动服务以及由该消息中的关联值标识的流程实例。如果指定了流程实例标识，那么将对使用关联值找到的流程实例进行检查，以确保它具有指定的流程实例标识。

处理事件

整个业务流程及其每个作用域都可以有相关联的事件处理程序，当相关联的事件发生时，将调用那些事件处理程序。就流程可以使用事件处理程序来提供 Web Service 操作而言，事件处理程序与接收或检取活动类似。

关于此任务

只要相应的作用域处于运行状态，就可以多次调用事件处理程序。并且，可以同时激活事件处理程序的多个实例。

以下代码段说明如何获取给定流程实例的活动事件处理程序以及如何发送输入消息。

过程

1. 确定流程实例标识的数据并列示该流程的活动事件处理程序。

```
ProcessInstanceData processInstance =
    process.getProcessInstance( "CustomerOrder2711");
EventHandlerTemplateData[] events = process.getActiveEventHandlers(
    processInstance.getID() );
```

2. 发送输入消息。

本示例使用找到的第一个事件处理程序。

```
EventHandlerTemplateData event = null;
if ( events.length > 0 )
{
    event = events[0];

    // create a message for the service to be called
    ClientObjectWrapper input = process.createMessage(
        event.getID(), event.getInputMessageType());

    if (input.getObject() != null && input.getObject() instanceof DataObject )
    {
        DataObject inputMessage = (DataObject)input.getObject();
        // set content of the message, for example, a customer name, order number
        inputMessage.setString("CustomerName", "Smith");
        inputMessage.setString("OrderNo", "2711");

        // send the message
        process.sendMessage( event.getProcessTemplateName(),
            event.getPortTypeNamespace(),
            event.getPortTypeName(),
            event.getOperationName(),

            input );
    }
}
```

此操作将指定的消息发送至流程的活动事件处理程序。

分析流程结果

流程可以发布作为 Web 服务描述语言（WSDL）单向操作或请求/响应操作设计的 Web Service 操作。由于具有单向接口的长时间运行流程没有输出，因此您无法使用 `getOutputMessage` 方法检索其结果。但是，可以查询变量的内容。

关于此任务

仅当用于派生流程实例的流程模板未指定自动删除所派生的流程实例时，流程结果才会存储在数据库中。

过程

分析流程结果，例如，检查订单号。

```
QueryResultSet result = process.query
    ("PROCESS_INSTANCE.PIID",
     "PROCESS_INSTANCE.NAME = 'CustomerOrder' AND
     PROCESS_INSTANCE.STATE =
     PROCESS_INSTANCE.STATE.STATE_FINISHED",
     (String)null, (Integer)null, (TimeZone)null);

if (result.size() > 0)
{
    result.first();
    PIID piid = (PIID) result.getOID(1);
    ClientObjectWrapper output = process.getOutputMessage(piid);
    DataObject myOutput = null;
    if ( output.getObject() != null && output.getObject() instanceof DataObject )
    {
        myOutput = (DataObject)output.getObject();
        int order = myOutput.getInt("OrderNo");
    }
}
```

修复活动

长时间运行流程可以包含运行时间同样较长的活动。这些活动可能会遇到未捕获的错误并进入“已停止”状态。处于运行状态的活动也可能会表现为未响应。在这两种情况下，流程管理员都可以通过许多方法对该活动执行操作，以使该流程的导航能够继续进行。

关于此任务

业务流程编排器 API 提供了 `forceRetry` 和 `forceComplete` 方法来修复活动。所提供的示例说明了如何在应用程序中添加活动修复操作。

强制完成活动

关于此任务

长时间运行流程中的活动有时会遇到故障。如果外层作用域中的故障处理程序未捕获这些故障，并且相关联的活动模块指定该活动在出错时应该停止，那么该活动将进入“已停止”状态以便进行修复。在此状态下，可以强制完成该活动。

对于处于“正在运行”状态的活动而言，如果该活动未响应，也可以强制使其完成。

某些类型的活动有一些其他要求。

人员任务活动

可以在强制完成调用中传递参数，例如应该已发送的消息或者应该已发生的故障。

脚本活动

不能在强制完成调用中传递参数。但是，必须设置需要修复的变量。

调用活动

如果调用活动调用除子流程以外的异步服务，并且这些活动处于“正在运行”状

态，那么也可以强制完成这些活动。例如，如果该异步服务被调用并且未作出响应，那么您将希望强制完成调用活动。

过程

1. 列示处于“已停止”状态的活动。

```
QueryResultSet result =
    process.query("DISTINCT ACTIVITY.AIID",
        "ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED AND
        PROCESS_INSTANCE.NAME='CustomerOrder'",
        (String)null, (Integer)null, (TimeZone)null);
```

此操作将返回 CustomerOrder 流程实例的已停止活动。

2. 完成活动，例如，完成已停止的人员任务活动。

在本示例中，传递了输出消息。

```
if (result.size() > 0)
{
    result.first();
    AIID aaid = (AIID) result.getOID(1);
    ActivityInstanceData activity = process.getActivityInstance(aaid);
    ClientObjectWrapper output =
        process.createMessage(aaid, activity.getOutputMessageType());
    DataObject myMessage = null;
    if ( output.getObject() != null && output.getObject() instanceof DataObject )
    {
        myMessage = (DataObject)output.getObject();
        //set the parts in your message, for example, an order number
        myMessage.setInt("OrderNo", 4711);
    }

    boolean continueOnError = true;
    process.forceComplete(aaid, output, continueOnError);
}
```

此操作将完成该活动。发生错误时，**continueOnError** 参数确定 forceComplete 请求附带提供了故障时要执行的操作。

在本示例中，**continueOnError** 值是 true。即，如果提供了故障，那么使该活动进入“已失败”状态。此故障将传播到该活动的外层作用域，直到它被处理或者到达流程作用域为止。如果该故障到达流程作用域，那么该流程将进入“即将失败”状态并最终进入“已失败”状态。

再次尝试执行已停止的活动

关于此任务

如果长时间运行流程中的活动在外层作用域中遇到未捕获的故障，并且相关联的活动模块指定该活动在出错时应该停止，那么该活动将进入“已停止”状态以便进行修复。您可以重新尝试执行该活动。

您可以设置该活动使用的变量。还可以在强制重试调用中传递参数，例如该活动所需的消息，但这不适用于脚本活动。

过程

1. 列示已停止的活动。

```

QueryResultSet result =
    process.query("DISTINCT ACTIVITY.AIID",
        "ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED AND
        PROCESS_INSTANCE.NAME='CustomerOrder'",
        (String)null, (Integer)null, (TimeZone)null);

```

此操作将返回 CustomerOrder 流程实例的已停止活动。

2. 再次尝试执行该活动，例如，执行已停止的人员任务活动。

```

if (result.size() > 0)
{
    result.first();
    AIID aaid = (AIID) result.getOID(1);
    ActivityInstanceData activity = process.getActivityInstance(aaid);
    ClientObjectWrapper input =
        process.createMessage(aaid, activity.getOutputMessageType());
    DataObject myMessage = null;
    if ( input.getObject() != null && input.getObject() instanceof DataObject )
    {
        myMessage = (DataObject)input.getObject();
        //set the strings in your message, for example, chocolate is to be ordered
        myMessage.setString("OrderNo", "chocolate");
    }

    boolean continueOnError = true;
    process.forceRetry(aaid, input, continueOnError);
}

```

此操作将重试该活动。**continueOnError** 参数确定了 forceRetry 请求处理期间出错时执行的操作。

在本示例中，**continueOnError** 值是 true。此值表示如果 forceRetry 请求处理期间出错，那么该活动将进入“已失败”状态。此故障将传播到该活动的外层作用域，直到它被处理或者到达流程作用域为止。该流程接着将进入“即将失败”状态，并且，在该流程最终进入“已失败”状态前，将运行流程级的故障处理程序。

BusinessFlowManagerService 接口

BusinessFlowManagerService 接口发布可以由客户机应用程序调用的业务流程函数。

可以由 BusinessFlowManagerService 接口调用的方法取决于流程状态或活动状态，并取决于包含该方法的应用程序的用户所具有的权限。下面列示了业务流程对象的主要处理方法。有关这些方法以及 BusinessFlowManagerService 接口中提供的其他方法的更多信息，请参阅 com.ibm.bpe.api 包中的 Javadoc。

流程模板

流程模板是已版本化、已部署并且已安装的流程模型，它包含业务流程的规范。可以通过发出诸如 sendMessage() 之类的适当请求将流程模板实例化以及将其启动。流程实例的执行由服务器自动驱动。

表 38. 流程模板的 API 方法

方法	描述
getProcessTemplate	检索指定的流程模板。
queryProcessTemplates	检索数据库中存储的流程模板。

流程实例

下列 API 方法与启动流程实例相关。

表 39. 与启动流程实例相关的 API 方法

方法	描述
call	创建并运行微流程。
callWithReplyContext	根据指定的流程模板创建并运行具有唯一启动服务的微流程或者具有唯一启动服务的长时间运行流程。该调用将以异步方式等待结果。
callWithUISettings	创建并运行微流程，然后返回输出消息和客户机用户界面（UI）设置。
initiate	创建流程实例并启动该流程实例的处理。此方法适用于长时间运行流程。也可以对要触发并忘记的微流程使用此方法。
sendMessage	将指定的消息发送至指定的活动服务和流程实例。如果不存在具有相同关联集值的流程实例，那么将创建该流程实例。该流程可以具有独特的启动服务，也可以具有不独特的启动服务。
getStartActivities	返回有关可以根据所指定流程模板启动流程实例的活动的信息。
getActivityServiceTemplate	检索指定的活动服务模板。

表 40. 用于控制流程实例生命周期的 API 方法

方法	描述
suspend	暂停执行处于“正在运行”或“即将失败”状态的顶层长时间运行流程实例。
resume	恢复处于“已暂挂”状态的顶层长时间运行流程实例。
restart	重新启动处于“已完成”、“已失败”或“已终止”状态的顶层长时间运行流程实例。
forceTerminate	终止指定的顶层流程实例及其 autonomy 属性值为 child 的子流程以及处于“正在运行”、“已声明”或“正在等待”状态的活动。
delete	删除指定的顶层流程实例及其 autonomy 属性值为 child 的子流程。
query	从数据库中检索与搜索条件匹配的属性。

活动

对于调用活动而言，可以在流程模型中指定这些活动在出错时将继续执行。如果将 `continueOnError` 标志设置为 `false`，并且发生了未处理的错误，该活动将进入“已停止”状态。于是，流程管理员可以修复该活动。例如，在调用活动偶尔失败但补偿建模和故障处理成本过高的长时间运行流程中，可以使用 `continueOnError` 标志和相关联的修复函数。

下列方法可用于处理和修复活动。

表 41. 用于控制活动实例生命周期的 API 方法

方法	描述
claim	声明已就绪的活动实例，以使用户处理该活动。
cancelClaim	取消声明活动实例。
complete	完成活动实例。
completeAndClaimSuccessor	完成活动实例并在登录用户的流程实例中声明下一个活动实例。
forceComplete	强制完成处于“正在运行”或“已停止”状态的活动实例。
forceRetry	强制重复处于“正在运行”或“已停止”状态的活动实例。
query	从数据库中检索与搜索条件匹配的属性。

变量和定制属性

接口提供了用于检索和设置变量值的 `get` 和 `set` 方法。还可以使指定的属性与流程实例和活动实例相关联，并可以从这些实例中检索指定的属性。定制属性名和属性值的类型必须是 `java.lang.String`。

表 42. 变量和定制属性的 API 方法

方法	描述
getVariable	检索指定的变量。
setVariable	设置指定的变量。
getCustomProperty	从指定的活动实例或流程实例中检索指定的定制属性。
getCustomProperties	从指定的活动实例或流程实例中检索多个定制属性。
getCustomPropertyNames	从指定的活动实例或流程实例中检索定制属性的名称。
setCustomProperty	为指定的活动实例或流程实例存储特定定制属性的值。

为人员任务开发应用程序

任务是组件以服务形式调用人员或者人员调用服务时采用的方法。本书提供人员任务的典型应用程序。

关于此任务

有关人员任务管理器 API 的更多信息，请参阅 `com.ibm.task.api` 包中的 Javadoc。

启动调用了同步接口的调用任务

调用任务与服务组件体系结构 (SCA) 组件相关联。该任务启动时，它将调用 SCA 组件。仅当能够以同步方式调用相关 SCA 组件时，才会以同步方式启动调用任务。

关于此任务

例如，这样的 SCA 组件可以作为微流程或简单 Java 类实现。

本方案创建任务模板的实例并传递一些客户数据。该任务在双向操作返回前将一直处于“正在运行”状态。该任务的结果 OrderNo 将返回给调用者。

过程

1. 可选： 列示任务模板，以查找所要运行的调用任务的名称。

如果您已知道该任务的名称，那么此步骤是可选的。

```
TaskTemplate[] taskTemplates = task.queryTaskTemplates(
    "TASK_TEMPL.KIND=TASK_TEMPL.KIND.KIND_ORIGINATING",
    "TASK_TEMPL.NAME",
    new Integer(50),
    (TimeZone)null);
```

结果将按名称排序。此查询返回一个数组，该数组包含前 50 个发端模板并已进行排序。

2. 创建适当类型的输入消息。

```
TaskTemplate template = taskTemplates[0];

// create a message for the selected task
ClientObjectWrapper input = task.createInputMessage( template.getID());
DataObject myMessage = null ;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
    myMessage = (DataObject)input.getObject();
    //set the parts in the message, for example, a customer name
    myMessage.setString("CustomerName", "Smith");
}
```

3. 创建该任务并以同步方式运行该任务。

要以同步方式运行任务，它必须是双向操作。本示例使用 createAndCallTask 方法来创建和运行任务。

```
ClientObjectWrapper output = task.createAndCallTask( template.getName(),
                                                    template.getNamespace(),
                                                    input);
```

4. 分析任务结果。

```
DataObject myOutput = null;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
    myOutput = (DataObject)output.getObject();
    int order = myOutput.getInt("OrderNo");
}
```

启动调用了异步接口的调用任务

调用任务与服务组件体系结构（SCA）组件相关联。该任务启动时，它将调用 SCA 组件。仅当能够以异步方式调用相关 SCA 组件时，才会以异步方式启动调用任务。

关于此任务

例如，这样的 SCA 组件可以作为长时间运行流程或单向操作实现。

本方案创建任务模板的实例并传递一些客户数据。

过程

1. 可选: 列示任务模板, 以查找所要运行的调用任务的名称。

如果您已知道该任务的名称, 那么此步骤是可选的。

```
TaskTemplate[] taskTemplates = task.queryTaskTemplates
("TASK_TEMPL.KIND=TASK_TEMPL.KIND.KIND_ORIGINATING",
 "TASK_TEMPL.NAME",
 new Integer(50),
 (TimeZone)null);
```

结果将按名称排序。此查询返回一个数组, 该数组包含前 50 个发端模板并已进行排序。

2. 创建适当类型的输入消息。

```
TaskTemplate template = taskTemplates[0];

// create a message for the selected task
ClientObjectWrapper input = task.createInputMessage( template.getID());
DataObject myMessage = null ;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
{
    myMessage = (DataObject)input.getObject();
    //set the parts in the message, for example, a customer name
    myMessage.setString("CustomerName", "Smith");
}
```

3. 创建该任务并以异步方式运行该任务。

本示例使用 `createAndStartTask` 方法来创建和运行任务。

```
task.createAndStartTask( template.getName(),
                        template.getNamespace(),
                        input,
                        (ReplyHandlerWrapper)null);
```

创建和启动任务实例

本方案说明如何创建定义了协作任务（在 API 中也称为人员任务）的任务模板实例以及如何启动该任务实例。

过程

1. 可选: 列示任务模板以查找所要运行的协作任务的名称。

如果您已知道该任务的名称, 那么此步骤是可选的。

```
TaskTemplate[] taskTemplates = task.queryTaskTemplates
("TASK_TEMPL.KIND=TASK_TEMPL.KIND.KIND_HUMAN",
 "TASK_TEMPL.NAME",
 new Integer(50),
 (TimeZone)null);
```

结果将按名称排序。此查询返回一个数组, 该数组包含前 50 个任务模板并已进行排序。

2. 创建适当类型的输入消息。

```
TaskTemplate template = taskTemplates[0];

// create a message for the selected task
ClientObjectWrapper input = task.createInputMessage( template.getID());
DataObject myMessage = null ;
if ( input.getObject() != null && input.getObject() instanceof DataObject )
```

```

{
  myMessage = (DataObject)input.getObject();
  //set the parts in the message, for example, a customer name
  myMessage.setString("CustomerName", "Smith");
}

```

3. 创建并启动该协作任务；在本示例中未指定应答处理程序。

本示例使用 `createAndStartTask` 方法来创建和启动任务。

```

TKIID tkiid = task.createAndStartTask( template.getName(),
                                     template.getNamespace(),
                                     input,
                                     (ReplyHandlerWrapper)null);

```

为该任务实例的相关人员创建了工作项。例如，潜在所有者可以声明新的任务实例。

4. 声明任务实例。

```

ClientObjectWrapper input2 = task.claim(tkiid);
DataObject taskInput = null ;
if ( input2.getObject() != null && input2.getObject() instanceof DataObject )
{
  taskInput = (DataObject)input2.getObject();
  // read the values
  ...
}

```

声明任务实例时，将返回该任务的输入消息。

处理待执行任务或协作任务

待执行任务（在 API 中也称为参与任务）或协作任务（在 API 中也称为人员任务）。通过工作项被分配给公司内的各种人员。例如，当流程进行到人员任务活动时，将创建待执行任务及其相关工作项。

关于此任务

其中一个潜在所有者声明与该工作项相关联的任务。此人员负责提供相关信息并完成该任务。

过程

1. 列示已登录人员能够处理的任務。

```

QueryResultSet result =
  task.query("TASK.TKIID",
            "TASK.STATE = TASK.STATE.STATE_READY AND
            (TASK.KIND = TASK.KIND.KIND_PARTICIPATING OR
            TASK.KIND = TASK.KIND.KIND_HUMAN)AND
            WORK_ITEM.REASON =
            WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
            (String)null, (Integer)null, (TimeZone)null);

```

此操作返回一个查询结果集，该结果集包含已登录人员能够处理的任務。

2. 声明要处理的任務。

```

if (result.size() > 0)
{
  result.first();
  TKIID tkiid = (TKIID) result.getOID(1);
  ClientObjectWrapper input = task.claim(tkiid);
  DataObject taskInput = null ;
}

```

```

    if ( input.getObject() != null && input.getObject() instanceof DataObject )
    {
        taskInput = (DataObject)input.getObject();
        // read the values
        ...
    }
}

```

声明任务时，将返回该任务的输入消息。

3. 声明处理任务后，将完成该任务。

该任务可能成功完成，也可能以报告故障消息而结束。如果该任务成功，那么将传递输出消息。如果该任务不成功，那么将传递故障消息。必须为这些操作创建适当的消息。

a. 要成功地完成任务，请创建输出消息。

```

ClientObjectWrapper output =
    task.createOutputMessage(tkiid);
DataObject myMessage = null ;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
    myMessage = (DataObject)output.getObject();
    //set the parts in your message, for example, an order number
    myMessage.setInt("OrderNo", 4711);
}

//complete the task
task.complete(tkiid, output);

```

此操作将设置包含序号的输出消息。该任务将进入“已完成”状态。

b. 要在发生故障时完成任务，请创建故障消息。

```

//retrieve the faults modeled for the task
List faultNames = task.getFaultNames(tkiid);

//create a message of the appropriate type
ClientObjectWrapper myFault =
    task.createFaultMessage(tkiid, (String)faultNames.get(0));

// set the parts in your fault message, for example, an error number
DataObject myMessage = null ;
if ( myFault.getObject() != null && input.getObject() instanceof DataObject )
{
    myMessage = (DataObject)myFault.getObject();
    //set the parts in the message, for example, a customer name
    myMessage.setInt("error",1304);
}

task.complete(tkiid, (String)faultNames.get(0), myFault);

```

此操作设置包含错误代码的故障消息。该任务将进入“已失败”状态。

暂挂和恢复任务实例

您可以暂挂协作任务实例（在 API 中也称为人员任务）或待执行任务实例（在 API 中也称为参与任务）。

开始之前

任务实例可以处于“就绪”或“已声明”状态。可以将其升级。调用者必须是该任务实例的所有者、发起者或管理员。

关于此任务

可以将处于“正在运行”状态的任务实例暂挂。例如，通过执行此操作，可以收集完成该任务所需的信息。获得该信息后，就可以恢复该任务实例。

过程

1. 获取登录用户声明的任务列表。

```
QueryResultSet result = task.query("DISTINCT TASK.TKIID",
                                   "TASK.STATE = TASK.STATE.STATE_CLAIMED",
                                   (String)null,
                                   (Integer)null,
                                   (TimeZone)null);
```

此操作返回一个查询结果集，该结果集包含登录用户声明的任务列表。

2. 暂挂任务实例。

```
if (result.size() > 0)
{
    result.first();
    TKIID tkiid = (TKIID) result.getOID(1);
    task.suspend(tkiid);
}
```

此操作暂挂指定的任务实例。该任务实例将进入“已暂挂”状态。

3. 恢复该流程实例。

```
task.resume( tkiid );
```

此操作将使该任务实例进入它被暂挂前所处的状态。

分析任务结果

待执行任务（在 API 中也称为参与任务）或协作任务（在 API 中也称为人员任务）以异步方式运行。如果任务启动时指定了应答处理程序，那么该任务完成时将自动返回输出消息。如果未指定应答处理程序，那么必须明确地检索该消息。

关于此任务

仅当用于派生任务实例的任务模板未指定自动删除所派生的任务实例时，任务结果才会存储在数据库中。

过程

分析任务结果。

本示例说明如何检查成功完成的任务的订单号。

```
QueryResultSet result = task.query("DISTINCT TASK.TKIID",
                                   "TASK.NAME = 'CustomerOrder' AND
                                   TASK.STATE = TASK.STATE.STATE_FINISHED",
                                   (String)null, (Integer)null, (TimeZone)null);

if (result.size() > 0)
{
    result.first();
    TKIID tkiid = (TKIID) result.getOID(1);
    ClientObjectWrapper output = task.getOutputMessage(tkiid);
    DataObject myOutput = null;
    if ( output.getObject() != null && output.getObject() instanceof DataObject)
    {
```

```

        myOutput = (DataObject)output.getObject();
        int order = myOutput.getInt("OrderNo");
    }
}

```

终止任务实例

有时，具有管理员权限的用户需要终止已知处于不可恢复状态的任务实例。由于任务实例将立即终止，因此，只有在异常情况下才应该终止任务实例。

过程

1. 检索要终止的任务实例。

```
Task taskInstance = task.getTask(tkiid);
```

2. 终止该任务实例。

```
TKIID tkiid = taskInstance.getID();
task.terminate(tkiid);
```

该任务实例将立即终止，而不会等待任何未完成的任务完成。

删除任务实例

仅当用于派生任务实例的相关任务模板指定应该自动删除任务实例时，任务实例在完成时才会自动被删除。以下示例说明如何删除所有已完成并且未被自动删除的任务实例。

过程

1. 列示已完成的任务实例。

```
QueryResultSet result =
    task.query("DISTINCT TASK.TKIID",
              "TASK.STATE = TASK.STATE.STATE_FINISHED",
              (String)null, (Integer)null, (TimeZone)null);
```

此操作将返回一个查询结果集，该结果集列示了已完成的任务实例。

2. 删除已完成的任务实例。

```
while (result.next() )
{
    TKIID tkiid = (TKIID) result.getOID(1);
    task.delete(tkiid);
}
```

释放已声明的任务

潜在所有者声明任务后，此人员将负责完成该任务。但是，有时必须将所声明的任务释放，以使另一个潜在所有者能够声明该任务。

关于此任务

有时，具有管理员权限的用户需要释放已声明的任务。例如，当必须完成某个任务，但该任务的所有者不在场时，将发生这种情况。已声明的任务也可以由该任务的所有者释放。

过程

1. 列示特定人员（例如 Smith）拥有的已声明任务。

```

QueryResultSet result =
    task.query("DISTINCT TASK.TKIID",
              "TASK.STATE = TASK.STATE.STATE_CLAIMED AND
              TASK.OWNER = 'Smith'",
              (String)null, (Integer)null, (TimeZone)null);

```

此操作返回一个查询结果集，该结果集列示了由指定人员（Smith）声明的任务。

2. 释放已声明的任务。

```

if (result.size() > 0)
{
    result.first();
    TKIID tkiid = (TKIID) result.getOID(1);
    task.cancelClaim(tkiid, true);
}

```

此操作使该任务回到“就绪”状态，以使其能够被其他潜在所有者声明。原始所有者所设置的输出或故障数据将保留下来。

管理工作项

在活动实例或任务实例的生存期内，与该对象相关联的人员集合会有所变化，例如，变化原因可能包括人员度假、聘用了新员工或者需要以另一方式分配工作。为了适应这些变化，您可以开发应用程序来创建、删除和移交工作项。

关于此任务

工作项表示由于特定原因将一个对象分配给用户或一组用户。通常，该对象是人员任务活动实例、流程实例或任务实例。原因从用户对该对象具有的角色派生。由于一个用户可以通过不同的角色与一个对象相关联，并且将为这些角色中的每一个创建工作项，所以一个对象可以有多个工作项。例如，待执行任务实例可以同时有管理员、阅读者、编辑者和所有者工作项。

管理工作项时可以执行的操作取决于用户的角色，例如，管理员能够创建、删除和移交工作项，但任务所有者只能移交工作项。

• 创建工作项。

```

// query the task instance for which an additional
// administrator is to be specified
QueryResultSet result = task.query("TASK.TKIID",
                                   "TASK.NAME='CustomerOrder'",
                                   (String)null, (Integer)null,
                                   (TimeZone)null);

if ( result.size() > 0 )
{
    result.first();
    // create the work item
    task.createWorkItem((TKIID)(result.getOID(1)),
                       WorkItem.REASON_ADMINISTRATOR,"Smith");
}

```

此操作为具有管理员角色的用户 Smith 创建工作项。

• 删除工作项。

```

// query the task instance for which a work item is to be deleted
QueryResultSet result = task.query("TASK.TKIID",
                                   "TASK.NAME='CustomerOrder'",
                                   (String)null, (Integer)null,
                                   (TimeZone)null);

if ( result.size() > 0 )

```



```

{
    result.first();
    // delete the work item
    task.deleteWorkItem((TKIID)(result.getOID(1)),
                        WorkItem.REASON_READER,"Smith");
}

```

此操作为具有阅读者角色的用户 Smith 删除工作项。

- 移交工作项。

```

// query the task that is to be rescheduled
QueryResultSet result =
    task.query("DISTINCT TASK.TKIID",
              "TASK.NAME='CustomerOrder' AND
              TASK.STATE=TASK.STATE.STATE_READY AND
              WORK_ITEM.REASON=WORK_ITEM.REASON.REASON_POTENTIAL_OWNER AND
              WORK_ITEM.OWNER_ID='Miller'",
              (String)null, (Integer)null, (TimeZone)null);
if ( result.size() > 0 )
{
    result.first();
    // transfer the work item from user Miller to user Smith
    // so that Smith can work on the task
    task.transferWorkItem((TKIID)(result.getOID(1)),
                          WorkItem.REASON_POTENTIAL_OWNER,"Miller","Smith");
}

```

此操作将工作项移交用户 Smith 以使其能够处理该工作项。

在运行时创建任务模板和任务实例

通常，使用建模工具（例如 WebSphere Integration Developer）来构建任务模板。然后，将任务模板安装到 WebSphere Process Server 中并根据这些模板创建实例，例如使用业务流程编排器资源管理器来创建实例。但是，也可以在运行时创建人员任务实例/模板或参与任务实例/模板。

关于此任务

例如，如果部署应用程序时没有可用的任务定义、工作流程包含未知的任务或者要通过某个任务处理一组人员之间的某些临时协作，那么您可能想在运行时创建实例/模板。

如果要设计临时待执行任务或协作任务，那么可以创建 `com.ibm.task.api.TaskModel` 类的实例并使用它们来创建可复用任务模板或者直接创建一次性任务实例。要创建 `TaskModel` 类的实例，可以使用 `com.ibm.task.api.ClientTaskFactory` 工厂类中提供的一组工厂方法。在运行时进行的人员任务设计工作基于 Eclipse 建模框架（EMF）。

过程

1. 通过使用 `createResourceSet` 工厂方法，创建 `org.eclipse.emf.ecore.resource.ResourceSet`。
2. 可选：如果您打算使用复杂消息类型，那么可以使用通过工厂方法 `getXSDFactory()` 获取的 `org.eclipse.xsd.XSDFactory` 来定义这些类型，也可以使用 `loadXSDDSchema` 工厂方法直接导入现有的 XML 模式。

要使复杂类型可供 WebSphere Process Server 使用，请将它们作为企业应用程序的组成部分进行部署。

3. 创建或导入类型为 `javax.wsdl.Definition` 的 Web 服务定义语言（WSDL）定义。

可以使用 `createWSDLDefinition` 方法来创建新的 WSDL 定义。然后，可以对其添加端口类型和操作。此外，还可以使用 `loadWSDLDefinition` 工厂方法直接导入现有的 WSDL 定义。

4. 使用 `createTTask` 工厂方法来创建任务定义。

如果要添加或处理多个复杂的任务元素，那么可以使用通过 `getTaskFactory` 工厂方法检索的 `com.ibm.wbit.tel.TaskFactory` 类。

5. 使用 `createTaskModel` 工厂方法创建任务模型，并将步骤 1 中创建的资源束（此资源束聚集了您同时创建的所有其他工件）传递给此模型。
6. 可选：使用 `TaskModel` 的 `validate` 方法来验证模型。

结果

使用其中一个带有 **TaskModel** 参数的人员任务管理器 EJB API `create` 方法来创建可复用任务模板或者一次性任务实例。

相关概念

第 34 页的『任务模板』

人员任务模板包含使用 WebSphere Integration Developer 创建的已部署任务模型的定义，或者包含在运行时使用业务流程编排器 API 创建的任务的定义。

创建使用简单 Java 类型的运行时任务

本示例创建一个运行时任务，该任务的接口只使用简单 Java 类型（例如 String 对象）。

关于此任务

本示例只能在已装入资源的主调企业应用程序的上下文中运行。

过程

1. 访问 `ClientTaskFactory` 并创建资源集以包含新任务模型的定义。

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();
ResourceSet resourceSet = factory.createResourceSet();
```

2. 创建 WSDL 定义并添加操作描述。

```
// create the WSDL interface
Definition definition = factory.createWSDLDefinition
    ( resourceSet, new QName( "http://www.ibm.com/task/test/", "test" ) );
```

```
// create a port type
PortType portType = factory.createPortType( definition, "doItPT" );
```

```
// create an operation; the input and output messages are of type String:
// a fault message is not specified
Operation operation = factory.createOperation
    ( definition, portType, "doIt",
      new QName( "http://www.w3.org/2001/XMLSchema", "string" ),
      new QName( "http://www.w3.org/2001/XMLSchema", "string" ),
      (Map)null );
```

3. 创建新人员任务的 EMF 模型。

如果创建任务实例，那么不需要“生效日期”（UTCDate）。

```
TTask humanTask = factory.createTTask( resourceSet,
                                         TTaskKinds.HTASK_LITERAL,
                                         "TestTask",
                                         new UTCDate( "2005-01-01T00:00:00" ),
```

```
"http://www.ibm.com/task/test/",
portType,
operation );
```

此步骤使用缺省值来初始化任务模型的属性。

4. 修改人员任务模型的属性。

```
// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance( TBoolean, YES_LITERAL );

// retrieve the task factory to create or modify composite task elements
TaskFactory taskFactory = factory.getTaskFactory();

// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");

// create escalationReceiver and add verb
TEscalationReceiver escalationReceiver =
    taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);

// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);
```

5. 创建包含所有资源定义的任务模型。

```
TaskModel taskModel = ClientTaskFactory.createTaskModel( resourceSet );
```

6. 验证任务模型并更正找到的验证问题。

```
ValidationProblem[] validationProblems = taskModel.validate();
```

7. 创建运行时任务实例或模板。

使用 `HumanTaskManagerService` 接口来创建任务实例或任务模板。由于此应用程序仅使用简单 Java 类型，所以不需要指定应用程序名。

- 以下代码段创建任务实例:

```
atask.createTask( taskModel, (String)null, "HTM" );
```

- 以下代码段创建任务模板:

```
task.createTaskTemplate( taskModel, (String)null );
```

结果

如果创建了运行时任务实例，那么现在可以将其启动。如果创建了运行时任务模板，那么现在可以根据该模板创建任务实例。

创建使用复杂类型的运行时任务

本示例创建一个运行时任务，该任务的接口使用复杂类型。复杂类型是已定义的类型，即，在客户机的本地文件系统中，有一些 XSD 文件包含复杂类型的描述。

关于此任务

本示例只能在已装入资源的主调企业应用程序的上下文中运行。

过程

1. 访问 `ClientTaskFactory` 并创建资源集以包含新任务模型的定义。

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();
ResourceSet resourceSet = factory.createResourceSet();
```

2. 将复杂类型的 XSD 定义添加到该资源集中，以便定义操作时能够使用这些定义。

文件位置相对于代码的执行位置。

```
factory.loadXSDSchema( resourceSet, "InputBO.xsd" );
factory.loadXSDSchema( resourceSet, "OutputBO.xsd" );
```

3. 创建 WSDL 定义并添加操作描述。

```
// create the WSDL interface
Definition definition = factory.createWSDLDefinition
    ( resourceSet, new QName( "http://www.ibm.com/task/test/", "test" ) );
```

```
// create a port type
PortType portType = factory.createPortType( definition, "doItPT" );
```

```
// create an operation; the input message is an InputBO and
// the output message an OutputBO;
// a fault message is not specified
Operation operation = factory.createOperation
    ( definition, portType, "doIt",
      new QName( "http://Input", "InputBO" ),
      new QName( "http://Output", "OutputBO" ),
      (Map)null );
```

4. 创建新人员任务的 EMF 模型。

如果创建任务实例，那么不需要“生效日期”（UTCDate）。

```
TTask humanTask = factory.createTTask( resourceSet,
    TTaskKinds.HTASK_LITERAL,
    "TestTask",
    new UTCDate( "2005-01-01T00:00:00" ),
    "http://www.ibm.com/task/test/",
    portType,
    operation );
```

此步骤使用缺省值来初始化任务模型的属性。

5. 修改人员任务模型的属性。

```
// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance( TBoolean, YES_LITERAL );
```

```
// retrieve the task factory to create or modify composite task elements
TaskFactory taskFactory = factory.getTaskFactory();
```

```
// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");
```

```
// create escalationReceiver and add verb
TEscalationReceiver escalationReceiver =
    taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);
```

```
// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);
```

6. 创建包含所有资源定义的任务模型。

```
TaskModel taskModel = ClientTaskFactory.createTaskModel( resourceSet );
```

7. 验证任务模型并更正找到的验证问题。

```
ValidationProblem[] validationProblems = taskModel.validate();
```

8. 创建运行时任务实例或模板。

使用 `HumanTaskManagerService` 接口来创建任务实例或任务模板。您必须提供包含数据类型定义的应用程序的名称，这样才能访问那些数据类型定义。该应用程序还必须包含一个哑任务或哑流程，这样业务流程编排器才能装入该应用程序。

- 以下代码段创建任务实例:

```
task.createTask( taskModel, "BOapplication", "HTM" );
```

- 以下代码段创建任务模板:

```
task.createTaskTemplate( taskModel, "BOapplication" );
```

结果

如果创建了运行时任务实例，那么现在可以将其启动。如果创建了运行时任务模板，那么现在可以根据该模板创建任务实例。

创建使用现有接口的运行时任务

本示例创建一个运行时任务。该任务使用已定义的接口，即，在客户机的本地文件系统中，有一个文件包含该接口的描述。

关于此任务

本示例只能在已装入资源的主调企业应用程序的上下文中运行。

过程

1. 访问 `ClientTaskFactory` 并创建资源集以包含新任务模型的定义。

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();
ResourceSet resourceSet = factory.createResourceSet();
```

2. 访问 WSDL 定义和操作描述。

接口描述的位置相对于代码的执行位置。

```
Definition definition = factory.loadWSDLDefinition(
    resourceSet, "interface.wsdl" );
PortType portType = definition.getPortType(
    new QName( definition.getTargetNamespace(), "doItPT" ) );
Operation operation = portType.getOperation(
    "doIt", (String)null, (String)null);
```

3. 创建新人员任务的 EMF 模型。

如果创建任务实例，那么不需要“生效日期”（`UTCDate`）。

```
TTask humanTask = factory.createTTask( resourceSet,
    TTaskKinds.HTASK_LITERAL,
    "TestTask",
    new UTCDate( "2005-01-01T00:00:00" ),
    "http://www.ibm.com/task/test/",
    portType,
    operation );
```

此步骤使用缺省值来初始化任务模型的属性。

4. 修改人员任务模型的属性。

```
// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance( TBoolean, YES_LITERAL );
```

```
// retrieve the task factory to create or modify composite task elements
```

```

TaskFactory taskFactory = factory.getTaskFactory();

// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");

// create escalationReceiver and add verb
TEscalationReceiver escalationReceiver =
    taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);

// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);

```

5. 创建包含所有资源定义的任务模型。

```
TaskModel taskModel = ClientTaskFactory.createTaskModel( resourceSet );
```

6. 验证任务模型并更正找到的验证问题。

```
ValidationProblem[] validationProblems = taskModel.validate();
```

7. 创建运行时任务实例或模板。

使用 `HumanTaskManagerService` 接口来创建任务实例或任务模板。您必须提供包含数据类型定义的应用程序的名称，这样才能访问那些数据类型定义。该应用程序还必须包含一个哑任务或哑流程，这样业务流程编排器才能装入该应用程序。

- 以下代码段创建任务实例:

```
task.createTask( taskModel, "B0Application", "HTM" );
```

- 以下代码段创建任务模板:

```
task.createTaskTemplate( taskModel, "B0Application" );
```

结果

如果创建了运行时任务实例，那么现在可以将其启动。如果创建了运行时任务模板，那么现在可以根据该模板创建任务实例。

创建使用主调应用程序中的接口的运行时任务

本示例创建一个运行时任务，该任务使用主调应用程序中的接口。例如，该运行时任务是在业务流程的 Java 片段中创建的，并且使用流程应用程序中的接口。

关于此任务

本示例只能在已装入资源的主调企业应用程序的上下文中运行。

过程

1. 访问 `ClientTaskFactory` 并创建资源集以包含新任务模型的定义。

```
ClientTaskFactory factory = ClientTaskFactory.newInstance();
```

```
// specify the context class loader so that following resources are found
ResourceSet resourceSet = factory.createResourceSet
    ( Thread.currentThread().getContextClassLoader() );
```

2. 访问 WSDL 定义和操作描述。

指定外层包 JAR 文件中的路径。

```

Definition definition = factory.loadWSDLDefinition( resourceSet,
    "com/ibm/workflow/metaflow/interface.wsdl" );
PortType portType = definition.getPortType(
    new QName( definition.getTargetNamespace(), "doItPT" ) );
Operation operation = portType.getOperation
    ("doIt", (String)null, (String)null);

```

3. 创建新人员任务的 EMF 模型。

如果创建任务实例，那么不需要“生效日期”（UTCDate）。

```

TTask humanTask = factory.createTTask( resourceSet,
    TTaskKinds.HTASK_LITERAL,
    "TestTask",
    new UTCDate( "2005-01-01T00:00:00" ),
    "http://www.ibm.com/task/test/",
    portType,
    operation );

```

此步骤使用缺省值来初始化任务模型的属性。

4. 修改人员任务模型的属性。

```

// use the methods from the com.ibm.wbit.tel package, for example,
humanTask.setBusinessRelevance( TBoolean, YES_LITERAL );

// retrieve the task factory to create or modify composite task elements
TaskFactory taskFactory = factory.getTaskFactory();

// specify escalation settings
TVerb verb = taskFactory.createTVerb();
verb.setName("John");

// create escalationReceiver and add verb
TEscalationReceiver escalationReceiver =
    taskFactory.createTEscalationReceiver();
escalationReceiver.setVerb(verb);

// create escalation and add escalation receiver
TEscalation escalation = taskFactory.createTEscalation();
escalation.setEscalationReceiver(escalationReceiver);

```

5. 创建包含所有资源定义的任务模型。

```
TaskModel taskModel = ClientTaskFactory.createTaskModel( resourceSet );
```

6. 验证任务模型并更正找到的验证问题。

```
ValidationProblem[] validationProblems = taskModel.validate();
```

7. 创建运行时任务实例或模板。

使用 `HumanTaskManagerService` 接口来创建任务实例或任务模板。您必须提供包含数据类型定义的应用程序的名称，这样才能访问那些数据类型定义。

- 以下代码段创建任务实例：

```
task.createTask( taskModel, "WorkflowApplication", "HTM" );
```

- 以下代码段创建任务模板：

```
task.createTaskTemplate( taskModel, "WorkflowApplication" );
```

结果

如果创建了运行时任务实例，那么现在可以将其启动。如果创建了运行时任务模板，那么现在可以根据该模板创建任务实例。

HumanTaskManagerService 接口

HumanTaskManagerService 接口发布与任务相关的函数，这些函数可以由本地或远程客户机调用。

可以调用的方法取决于任务状态，并取决于包含该方法的应用程序的用户所具有的权限。下面列示了任务对象的主要处理方法。有关这些方法以及 HumanTaskManagerService 接口中提供的其他方法的更多信息，请参阅 com.ibm.task.api 包中的 Javadoc。

任务模板

下列方法可用于处理任务模板。

表 43. 任务模板的 API 方法

方法	描述
getTaskTemplate	检索指定的任务模板。
createAndCallTask	根据指定的任务模板来创建并运行任务实例，然后以同步方式等待结果。
createAndStartTask	根据指定的任务模板来创建并启动任务实例。
createTask	根据指定的任务模板来创建任务实例。
createInputMessage	为指定的任务模板创建输入消息。例如，创建可用于启动任务的消息。
queryTaskTemplates	检索数据库中存储的任务模板。

任务实例

下列方法可用于处理任务实例。

表 44. 任务实例的 API 方法

方法	描述
getTask	检索任务实例；该任务实例可以处于任何状态。
callTask	以同步方式启动调用任务。
startTask	启动已创建的任务。
suspend	暂挂协作任务或待执行任务。
resume	恢复协作任务或待执行任务。
terminate	终止指定的任务实例。如果调用任务已终止，那么此操作不会影响所调用的服务。
delete	删除指定的任务实例。
claim	声明任务，以便进行处理。
update	更新任务实例。
complete	完成任务实例。
cancelClaim	释放已声明的任务实例，以使其可供另一潜在所有者处理。
createWorkItem	为任务实例创建工作项。
transferWorkItem	将工作项移交指定的所有者。
deleteWorkItem	删除工作项。

升级

下列方法可用于处理升级。

表 45. 用于处理升级的 API 方法

方法	描述
getEscalation	检索指定的升级实例。

定制属性

任务、任务模板和升级都可以具有定制属性。接口提供了用于检索和设置定制属性值的 `get` 和 `set` 方法。还可以使指定的属性与任务实例相关联，并可以从这些实例中检索指定的属性。定制属性名和属性值的类型必须是 `java.lang.String`。下列方法对任务、任务模板和升级有效。

表 46. 变量和定制属性的 API 方法

方法	描述
getCustomProperty	从指定的任务实例中检索指定的定制属性。
getCustomProperties	从指定的任务实例中检索多个定制属性。
getCustomPropertyNames	从任务实例中检索定制属性的名称。
setCustomProperty	为指定的任务实例存储特定于定制属性的值。

任务允许的操作

可以对任务执行的操作取决于该任务是待执行任务、协作任务、调用任务还是管理任务。

并非所有这些类型的任务都能使用所有由 `HumanTaskManager` 接口提供的操作。下表描述了可以对每一类任务执行的操作。

操作	任务类型			
	待执行任务	协作任务	调用任务	管理任务
callTask			X	
cancelClaim	X	X ¹		
claim	X	X ¹		
complete	X	X ¹		X
completeWithFollowOnTask ⁴	X	X ¹		
completeWithFollowOnTask ⁵		X ³	X ³	
createFaultMessage	X	X	X	X
createInputMessage	X	X	X	X
createOutputMessage	X	X	X	X
createWorkItem	X	X ¹	X	X
delete	X ¹	X ¹	X	X ¹
deleteWorkItem	X	X ¹	X	X
getCustomProperty	X	X ¹	X	X
getDocumentation	X	X ¹	X	X

操作	任务类型			
	待执行任务	协作任务	调用任务	管理任务
getFaultNames	X	X ¹		
getFaultMessage	X	X ¹	X	
getInputMessage	X	X ¹	X	
getOutputMessage	X	X ¹	X	
getUsersInRole	X	X ¹	X	X
getTask	X	X ¹	X	X
getUISettings	X	X ¹	X	X
resume	X	X ¹		
setCustomProperty	X	X ¹	X	X
setFaultMessage	X	X ¹		
setOutputMessage	X	X ¹		
startTask	X ¹	X ¹	X	X
startTaskAsSubtask ⁶	X	X ¹		
startTaskAsSubtask ⁷		X ³	X ³	
suspend	X	X ¹		
suspendWithCancelClaim	X	X ¹		
terminate	X ¹	X ¹	X ¹	
transferWorkItem	X	X ¹	X	X
update	X	X ¹	X	X
<p>注:</p> <ol style="list-style-type: none"> 1. 仅适用于独立任务、临时任务和任务模板 2. 仅适用于独立任务、业务流程中的直接插入任务以及临时任务 3. 仅适用于独立任务和临时任务 4. 可以带有后续任务的任务类型 5. 可以用作后续任务的任务类型 6. 可以带有子任务的任务类型 7. 可以用作子任务的任务类型 				

为业务流程和人员任务开发应用程序

大多数业务流程方案都涉及人员。例如，在启动或管理流程或者执行人员任务活动时，业务流程要求进行人员交互。为了支持这些方案，需要使用业务流程管理器 API 和人员任务管理器 API。

关于此任务

要让人员参与业务流程方案，可以在业务流程中包括下列类型的任务：

- 直接插入调用任务（在 API 中，也称为发端任务）。

可以为每个接收活动、检取活动的每个 `onMessage` 元素以及事件处理程序的每个 `onEvent` 元素提供调用任务。于是，此任务将控制哪些人员有权启动流程或者与运行中的流程实例进行通信。

- 管理任务。

您可以提供管理任务，以指定哪些人员有权管理流程或者对失败的流程活动执行管理操作。

- 待执行任务（在 API 中，也称为参与任务）。

待执行任务用于实现人员任务活动。此类活动允许人员参与流程。

业务流程中的人员任务活动代表人员在业务流程方案中执行的待执行任务。您可以使用业务流程管理器 API 和人员任务管理器 API 来实现这些方案：

- 业务流程是所有属于该流程的活动的容器，这些活动包括由待执行任务代表的人员任务活动。创建流程实例时，将指定唯一的对象标识（PIID）。
- 在流程实例执行期间激活人员任务活动时，将创建由唯一对象标识（AIID）进行标识的活动实例。同时，还将创建由对象标识（TKIID）进行标识的直接插入待执行任务实例。人员任务活动与任务实例之间的关系通过对象标识实现：
 - 活动实例的待执行任务标识设置为相关待执行任务的 TKIID。
 - 任务实例的包含上下文标识设置为包含相关活动实例的流程实例的 PIID。
 - 任务实例的父上下文标识设置为相关活动实例的 AIID。
- 所有直接插入待执行任务实例的生命周期都由流程实例管理。当流程实例被删除时，任务实例也将被删除。换言之，所有将包含上下文标识设置为流程实例 PIID 的任务都会被自动删除。

确定可以启动的流程模板或活动

可以通过调用业务流程管理器 API 的 `call`、`initiate` 或 `sendMessage` 方法来启动业务流程。如果该流程只有一个启动活动，那么可以使用要求流程模板名作为参数的方法特征符。如果该流程有多个启动活动，那么必须明确指定启动活动。

关于此任务

设计者在设计业务流程时，可以指定只有一小部分用户能够根据流程模板创建流程实例。要实现此目标，可以使直接插入调用任务与该流程的启动活动相关联，也可以对该任务指定授权限制。只有作为任务潜在启动者或管理员的人员才有权创建该任务的实例，从而创建该流程模板的实例。

如果未使任何直接插入调用任务与启动活动相关联，或者未对该任务指定授权限制，那么每个用户都可以使用该启动活动来创建流程实例。

一个流程可以有多个启动活动，每个启动活动都通过不同的人员查询来查询潜在启动者或管理员。这意味着，可以授权一个用户使用活动 A 来启动流程，但不允许他使用活动 B 来启动该流程。

过程

1. 使用业务流程管理器 API 来创建处于“已启动”状态的当前流程模板版本的列表。

提示： `queryProcessTemplates` 方法仅排除尚未启动的应用程序中包含的流程模板。因此，如果使用此方法但不对其结果进行过滤，那么此方法将返回所有版本的流程模板，而不考虑它们所处的状态。

```

// current timestamp in UTC format, converted to yyyy-mm-ddThh:mm:ss
String now = (new UTCDate()).toXsdString();
String whereClause = "PROCESS_TEMPLATE.STATE =
                    PROCESS_TEMPLATE.STATE.STATE_STARTED AND
                    PROCESS_TEMPLATE.VALID_FROM =
                    (SELECT MAX(VALID_FROM) FROM PROCESS_TEMPLATE
                     WHERE NAME=PROCESS_TEMPLATE.NAME AND
                     VALID_FROM <= TS('" + now + "'))";

ProcessTemplateData[] processTemplates = process.queryProcessTemplates
    ( whereClause,
      "PROCESS_TEMPLATE.NAME",
      (Integer)null, (TimeZone)null);

```

结果将按流程模板名排序。

2. 创建流程模板列表以及用户有权使用的启动活动的列表。

流程模板列表包含那些具有单一启动活动的流程模板。这些活动或者未受保护，或者只有登录用户才有权启动这些活动。此外，您可能想收集可以由至少一个启动活动启动的流程模板。

提示： 流程管理员也可以启动流程实例。要获取模板的完整列表，还需要读取与流程模板相关联的管理任务模板，并检查登录用户是否为管理员。

```

List authorizedProcessTemplates = new ArrayList();
List authorizedActivityServiceTemplates = new ArrayList();

```

3. 确定每个流程模板的启动活动。

```

for( int i=0; i<processTemplates.length; i++ )
{
    ProcessTemplateData template = processTemplates[i];
    ActivityServiceTemplateData[] startActivities =
        process.getStartActivities(template.getID());
}

```

4. 对于每个启动活动，检索相关联的直接插入调用任务模板的标识。

```

for( int j=0; j<startActivities.length; j++ )
{
    ActivityServiceTemplateData activity = startActivities[j];
    TKTID tktid = activity.getTaskTemplateID();
}

```

- a. 如果不存在调用任务模板，那么表示该流程模板未受此启动活动保护。

在这种情况下，每个人都可以使用此启动活动来创建流程实例。

```

boolean isAuthorized = false;
if ( tktid == null )
{
    isAuthorized = true;
    authorizedActivityServiceTemplates.add(activity);
}

```

- b. 如果已存在调用任务模板，那么请使用人员任务管理器 API 来检查登录用户的权限。

在本示例中，登录用户是 Smith。登录用户必须是调用任务的潜在启动者或者管理员。

```

if ( tktid != null )
{
    isAuthorized =
        task.isUserInRole
            (tkid, "Smith", WorkItem.REASON_POTENTIAL_STARTER) ||
        task.isUserInRole(tktid, "Smith", WorkItem.REASON_ADMINISTRATOR);
}

```

```

        if ( isAuthorized )
        {
            authorizedActivityServiceTemplates.add(activity);
        }
    }
}

```

如果该用户具有指定的角色，或者未指定该角色的人员分配条件，那么 `isUserInRole` 方法将返回 `true` 值。

5. 检查是否只能使用该流程模板名来启动该流程。

```

if ( isAuthorized && startActivities.length == 1 )
{
    authorizedProcessTemplates.add(template);
}

```

6. 结束循环。

```

    } // end of loop for each activity service template
} // end of loop for each process template

```

处理包含人员任务的单一人员工作流程

某些工作流程仅由一位人员执行，例如，从网上书店订购书籍。本示例说明如何将用于订购书籍的操作序列作为一系列人员任务活动（待执行任务）实现。在处理此工作流程的过程中，同时使用了业务流程管理器 API 和人员任务管理器 API。

关于此任务

在网上书店，顾客需要完成一系列操作以订购书籍。此操作序列可以作为一系列人员任务活动（待执行任务）实现。如果顾客决定订购多本书籍，那么等同于声明下一个人员任务活动。有关任务序列的信息由业务流程管理器维护，而任务本身由人员任务管理器维护。

请将此示例与仅使用业务流程管理器 API 的示例进行比较。

过程

1. 使用业务流程管理器 API 来获取要处理的流程实例。

在本示例中，处理 `CustomerOrder` 流程的实例。

```

ProcessInstanceData processInstance =
    process.getProcessInstance("CustomerOrder");
String piid = processInstance.getID().toString();

```

2. 使用人员任务管理器 API 来查询所指定流程实例中包含的已就绪的待执行任务（参与任务）。

使用该任务的包含上下文标识来指定外层流程实例。对于单一的个人工作流程而言，此查询将返回与人员任务活动序列中第一个人员任务活动相关联的待执行任务。

```

//
// Query the list of to-do tasks that can be claimed by the logged-on user
// for the specified process instance
//
QueryResultSet result =
    task.query("DISTINCT TASK.TKIID",
        "TASK.CONTAINMENT_CTX_ID = ID(' + piid + ') AND
        TASK.STATE = TASK.STATE.STATE_READY AND

```

```
TASK.KIND = TASK.KIND.KIND_PARTICIPATING AND
WORK_ITEM.REASON = WORK_ITEM.REASON.REASON_POTENTIAL_OWNER",
(String)null, (Integer)null, (TimeZone)null);
```

3. 声明所返回的待执行任务。

```
if (result.size() > 0)
{
    result.first();
    TKIID tkiid = (TKIID) result.getOID(1);
    ClientObjectWrapper input = task.claim(tkiid);
    DataObject activityInput = null ;
    if ( input.getObject() != null && input.getObject() instanceof DataObject )
    {
        taskInput = (DataObject)input.getObject();
        // read the values
        ...
    }
}
```

声明任务时，将返回该任务的输入消息。

4. 确定与待执行任务相关联的人员任务活动。

可以使用下列其中一个方法使活动与它们的任务相关。

- task.getActivityID 方法:

```
AIID aiid = task.getActivityID(tkiid);
```

- 任务对象所包含的父上下文标识:

```
AIID aiid = null;
Task taskInstance = task.getTask(tkiid);

OID oid = taskInstance.getParentContextID();
if ( oid != null and oid instanceof AIID )
{
    aiid = (AIID)oid;
}
```

5. 处理该任务完成后，请使用业务流程管理器 API 来完成该任务及其相关人员任务活动，然后声明流程实例中的下一个人员任务活动。

要完成该人员任务活动，必须传递输出消息。创建输出消息时，必须指定消息类型名，以便包含消息定义。

```
ActivityInstanceData activity = process.getActivityInstance(aiid);
ClientObjectWrapper output =
    process.createMessage(aiid, activity.getOutputMessageType());
DataObject myMessage = null ;
if ( output.getObject() != null && output.getObject() instanceof DataObject )
{
    myMessage = (DataObject)output.getObject();
    //set the parts in your message, for example, an order number
    myMessage.setInt("OrderNo", 4711);
}

//complete the human task activity and its associated to-do task,
// and claim the next human task activity
CompleteAndClaimSuccessorResult successor =
    process.completeAndClaimSuccessor(aiid, output);
```

此操作将设置包含订单号的输出消息并声明序列中的下一个人员任务活动。如果对后续活动设置了 AutoClaim，并且可以沿多条路径进行导航，那么将声明所有后续活动并随机返回一个活动作为下一个活动。如果没有其他后续活动可分配给此用户，那么将返回 Null。

如果该流程包含可以使用的并行路径，并且这些路径包含人员任务活动，已登录用户是多个此类活动的潜在所有者，那么将会自动声明随机活动并将其作为下一个活动返回。

6. 处理下一个人员任务活动。

```
ClientObjectWrapper nextInput = successor.getInputMessage();
if ( nextInput.getObject() !=
    null && nextInput.getObject() instanceof DataObject )
{
    activityInput = (DataObject)input.getObject();
    // read the values
    ...
}

aiid = successor.getAIID();
```

7. 转到步骤 5，以完成此人员任务活动并检索下一个人员任务活动。

处理异常和故障

在 BPEL 流程的各个执行阶段，可能会发生故障。

关于此任务

业务流程执行语言（BPEL）的故障源于：

- Web Service 调用（Web 服务描述语言（WSDL）故障）
- 抛出活动
- 业务流程编排器识别的 BPEL 标准故障

系统提供了用于处理这些故障的机制。请使用下列其中一种机制来处理流程实例生成的故障：

- 将控制权传递至相应的故障处理程序
- 对该流程中先前的工作进行补偿
- 停止该流程并允许某人处理这种情况，即强制重试或强制完成

BPEL 流程还可以将故障返回至该流程提供的操作的调用者。可以将流程中的故障设计成具有故障名和故障数据的应答活动。这些故障将作为已校验的异常返回给 API 调用者。

如果 BPEL 流程未处理 BPEL 故障，或者发生了 API 异常，那么会将运行时异常返回给 API 调用者。例如，创建实例时，如果所基于的流程模型不存在，那么将发生 API 异常。

下列任务描述如何处理故障和异常。

处理 API 异常

关于此任务

如果 BusinessFlowManagerService 接口或 HumanTaskManagerService 接口中的方法未成功完成，那么将抛出指示了错误原因的异常。您可以明确地处理此异常以便向调用者提供指导。

但是，通常的做法是仅明确地处理一小部分异常并提供其他潜在异常的一般指导。所有特定异常都继承通用的 `ProcessException` 或 `TaskException`。最好的做法是使用终态的 `catch(ProcessException)` 或 `catch(TaskException)` 语句来捕获一般异常。由于此语句将考虑所有其他可能发生的异常，所以它有助于确保应用程序向上兼容。

检查为活动设置了哪个故障

过程

1. 列示处于“已失败”或“已停止”状态的任务活动。

```
QueryResultSet result =
    process.query("ACTIVITY.AIID",
        "(ACTIVITY.STATE = ACTIVITY.STATE.STATE_FAILED OR
         ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED) AND
         ACTIVITY.KIND=ACTIVITY.KIND.KIND_STAFF",
        (String)null, (Integer)null, (TimeZone)null);
```

此操作返回一个查询结果集，该结果集包含已失败或已停止的活动。

2. 读取故障的名称。

```
if (result.size() > 0)
{
    result.first();
    AIID aaid = (AIID) result.getOID(1);
    ClientObjectWrapper faultMessage = process.getFaultMessage(aaid);
    DataObject fault = null ;
    if ( faultMessage.getObject() != null && faultMessage.getObject()
        instanceof DataObject )
    {
        fault = (DataObject)faultMessage.getObject();
        Type type = fault.getType();
        String name = type.getName();
        String uri = type.getURI();
    }
}
```

这段代码返回故障名。也可以对已停止活动的未处理异常进行分析，而不检索故障名。

检查已停止的调用活动所发生的故障

关于此任务

活动引起故障后，可通过故障类型确定可用于修复该活动的操作。

过程

1. 列示处于“已停止”状态的人员任务活动。

```
QueryResultSet result =
    process.query("ACTIVITY.AIID",
        "ACTIVITY.STATE = ACTIVITY.STATE.STATE_STOPPED AND
         ACTIVITY.KIND=ACTIVITY.KIND.KIND_INVOKE",
        (String)null, (Integer)null, (TimeZone)null);
```

此操作返回一个查询结果集，该结果集包含已停止的调用活动。

2. 读取故障的名称。

```
if (result.size() > 0)
{
    result.first();
    AIID aaid = (AIID) result.getOID(1);
```

```
ActivityInstanceData activity = process.getActivityInstance(aiid);  
  
ProcessException excp = activity.getUnhandledException();  
if ( excp instanceof ApplicationFaultException )  
{  
    ApplicationFaultException fault = (ApplicationFaultException)excp;  
    String faultName = fault.getFaultName();  
}  
}
```

第 12 章 开发 Web Service API 客户机应用程序

您可以开发通过 Web Service API 访问业务流程应用程序和人员任务应用程序的客户机应用程序。

关于此任务

可以在任何 Web Service 客户机环境中开发客户机应用程序，这些环境包括 Java Web Service 和 Microsoft .NET。

简介: Web Service

Web Service 是基于 Web 的企业应用程序，它使用基于 XML 的开放标准和传输协议与客户机应用程序交换数据。Web Service 允许使用与语言和环境无关的编程模型。

Web Service 使用了下列核心技术:

- XML (可扩展标记语言)。XML 解决了数据独立性问题。您使用 XML 来描述数据以及在该数据与任何应用程序或编程语言之间进行映射。
- WSDL (Web 服务描述语言)。这种基于 XML 的语言用于创建底层应用程序的描述。此描述用作底层应用程序与其他支持 Web 的应用程序之间的接口，从而将应用程序转换成 Web Service。
- SOAP (简单对象访问协议)。SOAP 是 Web 的核心通信协议，大部分 Web Service 使用此协议来相互进行通信。

Web Service 组件与控制序列

多个客户端和服务器端组件参与了代表 Web Service 请求和响应的控制序列。

典型的控制序列如下所示。

1. 在客户端:
 - a. 用户提供的客户机应用程序发出对 Web Service 的请求。
 - b. 代理客户机将该服务请求包装在 SOAP 请求包络中。此代理客户机也由用户提供，但可以使用客户端实用程序自动生成。
 - c. 客户端开发基础结构将该请求转发至定义为 Web Service 端点的 URL。
2. 网络使用 HTTP 或 HTTPS 将该请求传输至 Web Service 端点。
3. 在服务器端:
 - a. 通用 Web Service API 接收该请求并对其进行解码。
 - b. 该请求或者由通用业务流程管理器或人员任务管理器组件直接处理，或者被转发至指定的业务流程或人员任务。
 - c. 返回的数据包装在 SOAP 响应包络中。
4. 网络使用 HTTP 或 HTTPS 将该响应传输到客户端环境。
5. 回到客户端:
 - a. 客户端开发基础结构对 SOAP 响应包络进行解包。

- b. 代理客户机从 SOAP 响应中抽取数据并将其传递给客户机应用程序。
- c. 客户机应用程序根据需要对返回的数据进行处理。

Web Service API 概述

Web Service API 使您能够开发使用 Web Service 的客户机应用程序，以访问业务流程编排器环境中运行的业务流程和人员任务。

业务流程编排器 Web Service API 提供了两个独立的 Web Service 接口（WSDL 端口类型）：

- 业务流程管理器 API。允许客户机应用程序与微流程和长时间运行流程进行交互，例如：
 - 创建流程模板和流程实例
 - 声明现有的流程
 - 按流程的标识对其进行查询

请参阅第 389 页的『为业务流程开发应用程序』以获取可能操作的完整列表。

- 人员任务管理器 API。允许客户机应用程序执行下列操作：
 - 创建和启动任务
 - 声明现有的任务
 - 完成任务
 - 按任务的标识对其进行查询
 - 查询一组任务

请参阅第 407 页的『为人员任务开发应用程序』以获取可能操作的完整列表。

客户机应用程序可以使用其中一个 Web Service 接口，也可以同时使用这两个接口。

示例

以下是一个客户机应用程序的可能大纲，此应用程序通过访问人员任务管理器 Web Service API 来处理特定人员任务：

1. 客户机应用程序对 WebSphere Process Server 发出 Web Service 调用 query，请求获取用户要处理的参与任务列表。
2. 参与任务列表在 SOAP/HTTP 响应包络中返回。
3. 然后，客户机应用程序发出 Web Service 调用 claim，以声明其中一个参与任务。
4. WebSphere Process Server 返回该任务的输入消息。
5. 客户机应用程序发出 Web Service 调用 complete，以完成该任务并发出输入消息或故障消息。

对业务流程和人员任务的要求

使用 WebSphere Integration Developer 开发的对业务流程编排器运行的业务流程和人员任务必须符合特定的规则，这样才能通过 Web Service API 对它们进行访问。

这些要求包括：

1. 必须使用针对基于 XML 的 RPC 的 Java API (JAX-RPC 1.1) 规范中定义的“document/literal 包装”样式来定义业务流程接口和人员任务接口。此样式是使用 WID 开发的所有业务流程和人员任务的缺省样式。
2. 业务流程和人员任务发出的 Web Service 操作故障消息必须包含由 XML 模式元素定义的单一 WSDL 消息部件。例如:

```
<wsdl:part name="myFault" element="myNamespace:myFaultElement"/>
```

相关信息



针对基于 XML 的 RPC 的 Java API (JAX-RPC) 下载页面



我应该使用哪种样式的 WSDL?

开发客户机应用程序

客户机应用程序开发流程由多个步骤组成。

过程

1. 确定客户机应用程序需要使用的 Web Service API: 业务流程管理器 API 和/或人员任务管理器 API。
2. 从 WebSphere Process Server 环境中导出所需的文件。此外,也可以从 WebSphere Process Server 客户机 CD 复制这些文件。
3. 在选择的客户机应用程序开发环境中,使用导出的工件来生成代理客户机。
4. 可选: 生成辅助控件类。如果客户机应用程序直接与 WebSphere 服务器上的具体流程或任务进行交互,那么需要辅助控件类。但是,如果客户机应用程序只执行诸如发出查询之类的常规任务,那么不需要辅助控件类。
5. 开发客户机应用程序代码。
6. 在客户机应用程序中添加必需的安全性机制。

复制工件

为了帮助创建客户机应用程序,必须从 WebSphere 环境复制多个工件。

可以通过两种方法来获取这些工件:

- 从 WebSphere Process Server 环境中发布并导出这些工件。
- 从 WebSphere Process Server 客户机 CD 复制文件。

从服务器环境中发布和导出工件

在可以开发需要访问 Web Service API 的客户机应用程序之前,必须从 WebSphere 服务器环境中发布和导出多个工件。

关于此任务

需要导出的工件包括:

- Web 服务定义语言 (WSDL) 文件。这些文件描述了构成 Web Service API 的端口类型和操作。

- XML 模式定义 (XSD) 文件。这些文件包含 WSDL 文件中的服务和方法所引用的数据类型定义。
- 其他用于描述业务对象的 WSDL 和 XSD 文件。业务对象描述 WebSphere 服务器上运行的具体业务流程或人员任务。仅当客户机应用程序需要通过 Web Service API 直接与具体业务流程或人员任务进行交互时，才需要这些附加文件。如果客户机应用程序只执行诸如发出查询之类的常规任务，那么不需要这些文件。

发布这些工件后，需要将它们复制到客户机编程环境，以便使用它们来生成代理客户机和辅助控件类。

相关任务

第 440 页的『从客户机 CD 复制文件』

WebSphere Process Server 客户机 CD 提供了访问 Web Service API 时所需的文件。

指定 Web Service 端点地址

Web Service 端点地址是客户机应用程序访问 Web Service API 时必须指定的 URL。端点地址将被写入为了生成客户机应用程序的代理客户机而导出的 WSDL 文件。

关于此任务

要使用的 Web Service 端点地址取决于 WebSphere 服务器配置：

- 方案 1. 单个 WebSphere 服务器。要指定的 WebSphere 端点地址是服务器的主机名和端口号，例如 **host1:9080**。
- 方案 2. 由多个服务器组成的 WebSphere 集群。要指定的 WebSphere 端点地址是用于管理 Web Service API 的服务器名称和端口，例如 **host2:9081**。
- 方案 3. 使用 Web 服务器作为前端。要指定的 WebSphere 端点地址是该 Web 服务器的主机名和端口，例如 **host:80**。

缺省情况下，Web Service 端点地址格式为 *protocol://host:port/context_root/fixed_path*。其中：

- *protocol* 是客户机应用程序与 WebSphere 服务器之间使用的通信协议。缺省协议是 HTTP。可以使用更安全的 HTTPS（基于 SSL 的 HTTP）协议。建议您使用 HTTPS。
- *host:port* 是访问管理 Web Service API 的主机时使用的主机名和端口号。根据 WebSphere 服务器配置的不同，这些值也会有所变化；例如，客户机应用程序是直接访问应用程序还是通过 Web 服务器前端进行访问。
- *context_root* 是您可以随意选择的上下文根的值。但是，选择的值在每个 WebSphere 单元中必须唯一。缺省值使用“node_server/cluster”作为后缀，以消除命名冲突的风险。
- 对于业务流程管理器 API 而言，*fixed_path* 的值为 */sca/com/ibm/bpe/api/BFMWS*；而对于人员任务管理器 API 而言，此值为 */sca/com/ibm/task/api/HTMWS*，并且不可修改。

Web Service 端点地址最初由您在业务流程容器或人员任务容器配置期间指定：

过程

1. 使用具有管理员权限的用户标识登录管理控制台。
2. 选择应用程序 → **SCA** 模块。

注：也可以选择应用程序 → 企业应用程序以显示所有可用企业应用程序的列表。

3. 对于业务流程容器，从 SCA 模块或应用程序列表中选择 **BPEContainer**；对于人员任务容器，选择 **TaskContainer**。
4. 从其他属性列表中选择提供 **HTTP 端点 URL 信息**。
5. 从列表选择一个缺省前缀，或者输入定制前缀。如果客户机应用程序将直接连接至管理 Web Service API 的应用程序服务器，那么请使用缺省前缀列表中的前缀。否则，请指定定制前缀。
6. 单击**应用**以将所选前缀复制到 SCA 模块。
7. 单击**确定**。URL 信息将保存到工作空间中。

结果

您可以在管理控制台中查看当前值。例如，对于业务流程容器，请选择**企业应用程序 → BPEContainer → 查看部署描述符**。

在导出的 WSDL 文件中，soap:address 元素的 location 属性包含所指定的 Web Service 端点地址。例如：

```
<wsdl:service name="BFMWSservice">
  <wsdl:port name="BFMWSport" binding="this:BFMWSbinding">
    <soap:address location=
      "https://myserver:9080/WebServicesAPIs/sca/com/ibm/bpe/api/BFMWS"/>
  </wsdl:port>
</wsdl:service>
```

相关概念

第 449 页的『添加安全性 (Java Web Service)』

必须通过在客户机应用程序中实现安全性机制对 Web Service 通信进行保护。

相关任务

第 457 页的『添加安全性 (.NET)』

通过在客户机应用程序中集成安全性机制，可以对 Web Service 通信进行保护。

『发布 WSDL 文件』

Web 服务定义语言 (WSDL) 文件包含通过 Web Service API 提供的所有操作的详细描述。业务流程管理器 Web Service API 和人员任务管理器 Web Service API 有各自的 WSDL 文件。您首先必须发布这些 WSDL 文件，然后将它们从 WebSphere 环境复制到开发环境。在开发环境中，将使用这些 WSDL 文件来生成代理客户机。

发布 WSDL 文件

Web 服务定义语言 (WSDL) 文件包含通过 Web Service API 提供的所有操作的详细描述。业务流程管理器 Web Service API 和人员任务管理器 Web Service API 有各自的 WSDL 文件。您首先必须发布这些 WSDL 文件，然后将它们从 WebSphere 环境复制到开发环境。在开发环境中，将使用这些 WSDL 文件来生成代理客户机。

开始之前

在发布 WSDL 文件之前，务必指定正确的 Web Service 端点地址。此地址是一个 URL，客户机应用程序将使用此 URL 来访问 Web Service API。

关于此任务

WSDL 文件只需发布一次。

注: 如果您有 WebSphere Process Server 客户机 CD, 那么可以将这些文件从 CD 直接复制到客户机编程环境。

相关任务

第 453 页的『生成代理客户机 (.NET)』

.NET 客户机应用程序使用代理客户机与 Web Service API 进行交互。代理客户机使客户机应用程序不必处理复杂的 Web Service 消息传递协议。

第 436 页的『指定 Web Service 端点地址』

Web Service 端点地址是客户机应用程序访问 Web Service API 时必须指定的 URL。端点地址将被写入为了生成客户机应用程序的代理客户机而导出的 WSDL 文件。

发布业务流程 WSDL:

可以使用管理控制台来发布 WSDL 文件。

过程

1. 使用具有管理员权限的用户标识登录管理控制台。
2. 选择应用程序 → SCA 模块。

注: 也可以选择应用程序 → 企业应用程序以显示所有可用企业应用程序的列表。

3. 从 SCA 模块或应用程序列表中选择 **BPEContainer** 应用程序。
4. 从其他属性的列表中选择发布 **WSDL** 文件。
5. 单击列表中的 zip 文件。
6. 在打开的下载文件窗口中, 单击保存。
7. 浏览到某个本地文件夹, 然后单击保存。

结果

导出的 zip 文件名为 BPEContainer_WSDLFiles.zip。这个 zip 文件包含一个用于描述 Web Service 的 WSDL 文件以及该 WSDL 文件中引用的任何 XSD 文件。

发布人员任务 WSDL:

可以使用管理控制台来发布 WSDL 文件。

过程

1. 使用具有管理员权限的用户标识登录管理控制台。
2. 选择应用程序 → SCA 模块。

注: 也可以选择应用程序 → 企业应用程序以显示所有可用企业应用程序的列表。

3. 从 SCA 模块或应用程序列表中选择 **TaskContainer** 应用程序。
4. 从其他属性的列表中选择发布 **WSDL** 文件。
5. 单击列表中的 zip 文件。
6. 在打开的下载文件窗口中, 单击保存。
7. 浏览到某个本地文件夹, 然后单击保存。

结果

导出的 zip 文件名为 TaskContainer_WSDLFiles.zip。这个 zip 文件包含一个用于描述 Web Service 的 WSDL 文件以及该 WSDL 文件中引用的任何 XSD 文件。

导出业务对象

业务流程和人员任务具有明确的接口，通过这些接口，您可以将业务流程和人员任务作为 Web Service 从外部进行访问。如果这些接口引用了业务对象，那么需要将接口定义和业务对象导出到客户机编程环境。

关于此任务

必须对需要与客户机应用程序进行交互的每个业务对象重复本过程。

在 WebSphereProcess Server 中，业务对象定义了与业务流程或人员任务进行交互的请求、响应和故障消息的格式。这些消息还可以包含复杂数据类型的定义。

例如，要创建和启动人员任务，必须将下列信息项传递给任务接口：

- 任务模板名
- 任务模板名称空间
- 包含已格式化业务数据的输入消息
- 用于返回响应消息的响应包装器
- 用于返回故障和异常的故障消息

这些项封装在单个业务对象中。Web Service 接口的所有操作都是作为“document/literal 包装”操作设计的。这些操作的输入和输出参数包含在包装器文档中。其他业务对象定义了相应的响应和故障消息格式。

为了通过 Web Service 创建和启动业务流程或人员任务，必须向客户机端的客户机应用程序提供这些包装器对象。

实现方式如下所示：从 WebSphere 环境中将业务对象作为 Web 服务定义语言（WSDL）和 XML 模式定义（XSD）文件导出，将数据类型定义导入客户机编程环境，然后将它们转换为供客户机应用程序使用的辅助控件类。

过程

1. 如果 WebSphere Integration Developer 工作空间未处于运行状态，那么将其启动。
2. 选择包含要导出的业务对象的库模块。库模块是包含所需业务对象的压缩文件。
3. 导出库模块。
4. 将导出的文件复制到客户机应用程序开发环境。

示例

假定业务流程发布了下列 Web Service 操作：

```
<wsdl:operation name="updateCustomer">
  <wsdl:input message="tns:updateCustomerRequestMsg"
    name="updateCustomerRequest"/>
  <wsdl:output message="tns:updateCustomerResponseMsg"
    name="updateCustomerResponse"/>
  <wsdl:fault message="tns:updateCustomerFaultMsg"
    name="updateCustomerFault"/>
</wsdl:operation>
```

并且 WSDL 消息定义为:

```
<wsdl:message name="updateCustomerRequestMsg">
  <wsdl:part element="types:updateCustomer"
    name="updateCustomerParameters"/>
</wsdl:message>
<wsdl:message name="updateCustomerResponseMsg">
  <wsdl:part element="types:updateCustomerResponse"
    name="updateCustomerResult"/>
</wsdl:message>
<wsdl:message name="updateCustomerFaultMsg">
  <wsdl:part element="types:updateCustomerFault"
    name="updateCustomerFault"/>
</wsdl:message>
```

在客户机应用程序执行的 call 和 sendMessage 等所有通用操作中, 必须使用 <xsd:any> 参数将具体的客户定义元素 types:updateCustomer、types:updateCustomerResponse 和 types:updateCustomerFault 传递给 Web Service API 或者从 Web Service API 接收这些元素。这些客户定义元素在客户机应用程序端使用辅助控件类进行创建、序列化和反序列化, 而辅助控件类通过已导出的 XSD 文件生成。

相关任务

第 454 页的『创建 BPEL 流程的辅助控件类 (.NET)』

某些 Web Service API 操作要求客户机应用程序使用“document/literal”样式包装的元素。客户机应用程序需要使用辅助控件类来帮助生成必需的包装器元素。

第 454 页的『创建 BPEL 流程的辅助控件类 (.NET)』

某些 Web Service API 操作要求客户机应用程序使用“document/literal”样式包装的元素。客户机应用程序需要使用辅助控件类来帮助生成必需的包装器元素。

第 447 页的『创建 BPEL 流程的辅助控件类 (Java Web Service)』

在诸如 sendMessage 或 call 等具体 API 请求中引用的业务对象要求客户机应用程序使用“document/literal 包装”样式元素。客户机应用程序需要使用辅助控件类来帮助生成必需的包装器元素。

使用客户机 CD 上的文件

除了从 WebSphere 服务器环境中导出工件以外, 还可以从 WebSphere Process Server 客户机 CD 复制生成客户机应用程序时所必需文件。

在这种情况下, 必须手动修改业务流程管理器 API 或人员任务管理器 API 的缺省 Web Service 端点地址。

如果客户机应用程序要同时访问这两个 API, 那么您必须编辑这两个 API 的缺省端点地址。

从客户机 CD 复制文件

WebSphere Process Server 客户机 CD 提供了访问 Web Service API 时所需的文件。

过程

1. 访问客户机 CD 并转到 ProcessChoreographer\client 目录。
2. 将所需的文件复制到客户机应用程序开发环境。

对于业务流程管理器 API, 请复制:

BFMWS.wsdl

描述业务流程管理器 Web Service API 中提供的 Web Service。此文件包含端点地址。

BFMIF.wsdl

描述业务流程管理器 Web Service API 中每个 Web Service 的参数和数据类型。

BFMIF.xsd

描述业务流程管理器 Web Service API 中使用的数据类型。

BPCGEN.xsd

包含业务流程管理器 Web Service API 和人员任务管理器 Web Service API 的公共数据类型。

对于人员任务管理器 API，请复制：

HTMWS.wsdl

描述人员任务管理器 Web Service API 中提供的 Web Service。此文件包含端点地址。

HTMIF.wsdl

描述人员任务管理器 Web Service API 中每个 Web Service 的参数和数据类型。

HTMIF.xsd

描述人员任务管理器 Web Service API 中使用的数据类型。

BPCGEN.xsd

包含业务流程管理器 Web Service API 和人员任务管理器 Web Service API 的公共数据类型。

注： BPCGen.xsd 是这两个 API 的公共文件。

复制文件后，必须将 BFMWS.wsdl 或 HTMWS.wsdl 文件中的 Web Service API 端点地址手动更改为管理 Web Service API 的 WebSphere Application Server 的端点地址。

相关任务

『手动更改 Web Service 端点地址』

如果从客户机 CD 复制文件，那么必须将 WSDL 文件中指定的缺省 Web Service 端点地址更改为管理 Web Service API 的服务器的地址。

第 435 页的『从服务器环境中发布和导出工件』

在可以开发需要访问 Web Service API 的客户机应用程序之前，必须从 WebSphere 服务器环境中发布和导出多个工件。

手动更改 Web Service 端点地址

如果从客户机 CD 复制文件，那么必须将 WSDL 文件中指定的缺省 Web Service 端点地址更改为管理 Web Service API 的服务器的地址。

关于此任务

可以在导出 WSDL 文件前使用管理控制台来设置 Web Service 端点地址。但是，如果从 WebSphere Process Server 客户机 CD 复制 WSDL 文件，那么必须手动修改缺省 Web Service 端点地址。

要使用的 Web Service 端点地址取决于 WebSphere 服务器配置:

- 方案 1. 单个 WebSphere 服务器。要指定的 WebSphere 端点地址是服务器的主机名和端口号, 例如 **host1:9080**。
- 方案 2. 由多个服务器组成的 WebSphere 集群。要指定的 WebSphere 端点地址是用于管理 Web Service API 的服务器名称和端口, 例如 **host2:9081**。
- 方案 3. 使用 Web 服务器作为前端。要指定的 WebSphere 端点地址是该 Web 服务器的主机名和端口, 例如 **host:80**。

相关任务

第 440 页的『从客户机 CD 复制文件』

WebSphere Process Server 客户机 CD 提供了访问 Web Service API 时所需的文件。

更改业务流程管理器 API 端点:

如果从 WebSphere Process Server 客户机 CD 复制业务流程管理器 API 文件, 那么必须对缺省端点地址进行手动编辑。

过程

1. 转到从客户机 CD 复制的文件所在的目录。
2. 在文本编辑器或 XML 编辑器中打开 BFMWS.wsdl 文件。
3. 在此文件中, 从上到下查找 soap:address 元素。
4. 将 location 属性值更改为运行 Web Service API 的服务器的 HTTP URL。要完成此任务:
 - a. (可选) 将 http 替换为 https 以使用更安全的 HTTPS 协议。
 - b. 将 localhost 替换为 Web Service API 服务器端点地址的主机名或 IP 地址。
 - c. 将 9080 替换为应用程序服务器的端口号。
 - d. 将 *BPEContainer_N1_server1* 替换为运行 Web Service API 的应用程序的上下文根。缺省上下文根包含:
 - *BPEContainer*: 应用程序名。
 - *N1*: 节点名。
 - *server1*: 服务器名。
 - e. 不要修改 URL 的固定部分 (/sca/com/ibm/bpe/api/BFMWS)。

例如, 如果应用程序在服务器 **s1.n1.ibm.com** 上运行, 并且该服务器接受端口 **9080** 上的 SOAP/HTTP 请求, 那么将 soap:address 元素修改为:

```
<soap:address location="http://s1.n1.ibm.com:9080/  
BPEContainer_N1_server1/sca/com/ibm/bpe/api/BFMWS"/>
```

相关概念

第 449 页的『添加安全性 (Java Web Service)』

必须通过在客户机应用程序中实现安全性机制对 Web Service 通信进行保护。

相关任务

第 457 页的『添加安全性 (.NET)』

通过在客户机应用程序中集成安全性机制, 可以对 Web Service 通信进行保护。

更改人员任务管理器 API 端点:

如果从 WebSphere Process Server 客户机 CD 复制人员任务管理器 API 文件，那么必须对缺省端点地址进行手动编辑。

过程

1. 转到从客户机 CD 复制的文件所在的目录。
2. 在文本编辑器或 XML 编辑器中打开 HTMWS.wsdl 文件。
3. 在此文件中，从上到下查找 soap:address 元素。
4. 将 location 属性值更改为正确的端点地址。 要完成此任务：
 - a. (可选) 将 http 替换为 https 以使用更安全的 HTTPS 协议。
 - b. 将 localhost 替换为 Web Service API 服务器端点地址的主机名或 IP 地址。
 - c. 将 9080 替换为应用程序服务器的端口号。
 - d. 将 *HTMContainer_N1_server1* 替换为运行 Web Service API 的应用程序的上下文根。 缺省上下文根包含：
 - *HTMContainer*: 应用程序名。
 - *N1*: 节点名。
 - *server1*: 服务器名。
 - e. 不要修改 URL 的固定部分 (/sca/com/ibm/task/api/HTMWS)。

例如，如果应用程序在服务器 **s1.n1.ibm.com** 上运行，并且该服务器接受端口 **9081** 上的 SOAP/HTTPS 请求，那么将 soap:address 元素修改为：

```
<soap:address location="https://s1.n1.ibm.com:9081/  
HTMContainer_N1_server1/sca/com/ibm/task/api/HTMWS"/>
```

相关概念

第 449 页的『添加安全性 (Java Web Service)』

必须通过在客户机应用程序中实现安全性机制对 Web Service 通信进行保护。

相关任务

第 457 页的『添加安全性 (.NET)』

通过在客户机应用程序中集成安全性机制，可以对 Web Service 通信进行保护。

在 Java Web Service 环境中开发客户机应用程序

可以使用任何基于 Java 并与 Java Web Service 兼容的开发环境来开发 Web Service API 的客户机应用程序。

生成代理客户机 (Java Web Service)

Java Web Service 客户机应用程序使用代理客户机与 Web Service API 进行交互。

关于此任务

Java Web Service 的代理客户机包含多个 Java Bean 类，客户机应用程序通过调用这些 Bean 类来执行 Web Service 请求。代理客户机负责将服务参数组装成 SOAP 消息、通过 HTTP 将 SOAP 消息发送至 Web Service、从 Web Service 接收响应以及将返回的数据传递给客户机应用程序。

因此，代理客户机使客户机应用程序能够调用 Web Service，就象该 Web Service 是局部函数一样。

注：代理客户机只需生成一次。一旦生成，所有访问同一 Web Service API 的客户机应用程序都可以使用该代理客户机。

在 IBM Web Service 环境中，可以通过两种方法来生成代理客户机：

- 使用 Rational® Application Developer 或 WebSphere Integration Developer 集成开发环境。
- 使用 WSDL2Java 命令行工具。

其他 Java Web Service 开发环境通常包括 WSDL2Java 工具或专用的客户机应用程序生成工具。

使用 Rational Application Developer 生成代理客户机

Rational Application Developer 集成开发环境使您能够为客户机应用程序生成代理客户机。

开始之前

在生成代理客户机之前，必须已从 WebSphere 环境导出或者从 WebSphere Process Server 客户机 CD 复制用于描述业务流程或人员任务 Web Service 接口的 WSDL 文件并将这些文件复制到客户机编程环境。

过程

1. 将适当的 WSDL 文件添加到项目中：

- 对于业务流程：
 - a. 将导出的文件 BPEContainer_nodename_servername_WSDLFiles.zip 解压缩到一个临时目录中。
 - b. 从解压缩目录 BPEContainer_nodename_servername.ear/b.jar 中导入子目录 META-INF。
- 对于人员任务：
 - a. 将导出的文件 TaskContainer_nodename_servername_WSDLFiles.zip 解压缩到一个临时目录中。
 - b. 从解压缩目录 TaskContainer_nodename_servername.ear/h.jar 中导入子目录 META-INF。

在项目中将创建新目录 wsdl 和子目录结构。

2. 修改 Web Service 向导属性：

- a. 在 Rational Application Developer 中，选择首选项 → **Web Service** → **代码生成** → **IBM WebSphere 运行时**。
- b. 选中使用非包装样式根据 **WSDL 生成 Java** 选项。

注：如果无法从首选项菜单中选择 **Web Service** 选项，那么必须先启用必需的功能，如下所示：单击窗口 → 首选项 → 工作台 → 功能。单击 **Web Service 开发者**，然后单击**确定**。然后，重新打开首选项窗口并更改**代码生成**选项。

3. 从新创建的 wsdl 目录中选择 BFMWS.WSDL 或 HTMWWS.WSDL 文件。

4. 单击鼠标右键，然后选择 **Web Service** → **生成客户机**。

在继续执行余下的步骤前，请确保启动服务器。

5. 在 Web Service 窗口中，单击**下一步**以接受所有缺省值。
6. 在选择 Web Service 窗口中，单击**下一步**以接受所有缺省值。
7. 在客户机环境配置窗口中，执行下列操作：
 - a. 单击**编辑**并将 Web Service 运行时选项更改为 IBM WebSphere。
 - b. 将 J2EE 版本选项更改为 1.4。
 - c. 单击**确定**。
 - d. 单击**下一步**。
8. 仅当需要生成同时包含业务流程和人员任务 Web Service API 的 Web Service 客户机时，才必须执行此步骤，这是因为两个 WSDL 文件包含重复的方法。
 - a. 在“Web Service 代理”窗口中，选择定义名称空间到包的定制映射，然后单击**确定**。
 - b. 在“Web Service 客户机名称空间到包的映射”窗口中，添加下列名称空间和包：

对于 BFMWS.wsdl:

名称空间	包
http://www.ibm.com/xmlns/prod/websphere/business-process/types/6.0	com.ibm.sca.bpe
http://www.ibm.com/xmlns/prod/websphere/business-process/services/6.0	com.ibm.sca.bpe
http://www.ibm.com/xmlns/prod/websphere/business-process/services/6.0/ Binding	com.ibm.sca.bpe
http://www.ibm.com/xmlns/prod/websphere/bpc-common/types/6.0	com.ibm.sca.bpe

对于 HTMWWS.wsdl:

名称空间	包
http://www.ibm.com/xmlns/prod/websphere/human-task/types/6.0	com.ibm.sca.task
http://www.ibm.com/xmlns/prod/websphere/human-task/services/6.0	com.ibm.sca.task
http://www.ibm.com/xmlns/prod/websphere/human-task/services/6.0/ Binding	com.ibm.sca.task
http://www.ibm.com/xmlns/prod/websphere/bpc-common/types/6.0	com.ibm.sca.task

如果系统要求您确认覆盖，请单击**全部都是**。

9. 单击**完成**。

结果

将生成包含多个代理、定位器和辅助控件 Java 类的代理客户机并将其添加到项目中。还将更新部署描述符。

使用 WSDL2Java 生成代理客户机

WSDL2Java 是用于生成代理客户机的命令行工具。代理客户机简化了客户机应用程序的编程工作。

开始之前

在生成代理客户机之前，必须从 WebSphere 环境导出或者从 WebSphere Process Server 客户机 CD 复制用于描述业务流程或人员任务 Web Service API 的 WSDL 文件，然后将这些文件复制到客户机编程环境。

关于此任务

过程

1. 使用 WSDL2Java 工具来生成代理客户机： 输入：

```
wsdl2java options WSDLfilepath
```

其中：

- *options* 包括：

-noWrappedOperations (-w)

禁止检测所包装的操作。将生成请求消息和响应消息的 Java Bean。

注：这不是缺省值。

-role (-r)

指定值 **client** 以生成用于进行客户端开发的文件和绑定文件。

-container (-c)

要使用的客户端容器。有效参数包括：

client 客户机容器

ejb Enterprise JavaBeans (EJB) 容器

none 没有容器

web Web 容器

-output (-o)

用于存储所生成文件的文件夹。

要获取完整的 WSDL2Java 参数列表，请使用 **-help** 命令行开关，或者参考 WID/RAD 中 WSDL2Java 工具的联机帮助。

- *WSDLfilepath* 是从 WebSphere 环境导出的或者从客户机 CD 复制的 WSDL 文件的路径和文件名。

以下示例生成人员任务活动 Web Service API 的代理客户机：

```
call wsdl2java.bat -r client -c client -noWrappedOperations  
-output c:\ws\proxyClient c:\ws\bin\HTMWS.wsdl
```

2. 将所生成的类文件包括在项目中。

相关任务

第 448 页的『创建客户机应用程序 (Java Web Service)』

客户机应用程序将请求发送至 Web Service API 并从 Web Service API 接收响应。通过使用代理客户机来管理通信并使用辅助控件类来格式化复杂的数据类型，客户机应用程序可以调用 Web Service 方法，就象这些方法是局部函数一样。

创建 BPEL 流程的辅助控件类 (Java Web Service)

在诸如 `sendMessage` 或 `call` 等具体 API 请求中引用的业务对象要求客户机应用程序使用“document/literal 包装”样式元素。客户机应用程序需要使用辅助控件类来帮助生成必需的包装器元素。

开始之前

要创建辅助控件类，必须先从 WebSphere Process Server 环境中导出 Web Service API 的 WSDL 文件。

关于此任务

Web Service API 的 `call()` 和 `sendMessage()` 操作允许与 WebSphere Process Server 中的 BPEL 流程进行交互。`call()` 操作的输入消息要求您提供流程输入消息的 `document/literal` 包装器。

可以通过多种技术生成 BPEL 流程或人员任务的辅助控件类，其中包括：

1. 使用 SoapElement 对象。

在 WebSphere Integration Developer 中提供的 Rational Application Developer 环境中，Web Service 引擎支持 JAX-RPC 1.1。在 JAX-RPC 1.1 中，`SoapElement` 对象扩展文档对象模型 (DOM) 元素，因此，可以使用 DOM API 创建、读取、装入和保存 SOAP 消息。

例如，假定 WSDL 文件包含工作流过程或人员任务的以下输入消息：

```
<xsd:element name="operation1">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="input1" nillable="true" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

WSDL 文件是您开发流程或人员任务模块时创建的。

要使用 DOM API 在客户机应用程序中创建相应的 SOAP 消息：

```
SOAPFactory soapfactoryinstance = SOAPFactory.newInstance();
SOAPElement soapmessage = soapfactoryinstance.createElement
    ("operation1", namespaceprefix, interfaceURI);
SOAPElement inputelement = soapfactoryinstance.createElement("input1");
inputelement.addTextNode( message value);
soapmessage.addChildElement(outputelement);
```

以下示例说明如何在客户机应用程序中创建 `sendMessage` 操作的输入参数：

```
SendMessage inWsend = new SendMessage();
inWsend.setProcessTemplateName(processTemplateName);
inWsend.setPortType(portType);
inWsend.setOperation(operationName);
inWsend.set_any(soapmessage);
```

2. 使用 WebSphere 定制数据绑定功能。

下列 `developerWorks` 文章对此技术作了描述：

- 如何为 Web service 选择定制映射技术
- 使用 EMF SDO 为复杂的 XML 模式开发 Web Service

相关任务

第 439 页的『导出业务对象』

业务流程和人员任务具有明确的接口，通过这些接口，您可以将业务流程和人员任务作为 Web Service 从外部进行访问。如果这些接口引用了业务对象，那么需要将接口定义和业务对象导出到客户机编程环境。

 与基于文档的 Web service 的模式和策略的互操作性

 对包含可选 JAX-RPC 1.0/1.1 XML 模式类型的模式/WSDL 的 Web Service 支持

创建客户机应用程序 (Java Web Service)

客户机应用程序将请求发送至 Web Service API 并从 Web Service API 接收响应。通过使用代理客户机来管理通信并使用辅助控件类来格式化复杂的数据类型，客户机应用程序可以调用 Web Service 方法，就象这些方法是局部函数一样。

开始之前

在开始创建客户机应用程序前，请生成代理客户机以及任何所需的辅助控件类。

关于此任务

可以使用 IBM Rational Application Developer (RAD) 等与 Web Service 兼容的开发工具来开发客户机应用程序。可以构建任何类型的 Web Service 应用程序，以调用 Web Service API。

过程

1. 创建新的客户机应用程序项目。
2. 生成代理客户机，并将 Java 辅助控件类添加到项目中。
3. 编写客户机应用程序代码。
4. 构建项目。
5. 运行客户机应用程序。

以下示例说明如何使用业务流程管理器 Web Service API。

```
// create the proxy
    BFMIFProxy proxy = new BFMIFProxy();
// prepare the input data for the operation
    GetProcessTemplate iw = new GetProcessTemplate();
    iw.setIdentifier(your_process_template_name);

// invoke the operation
    GetProcessTemplateResponse oW = proxy.getProcessTemplate(iw);

// process output of the operation
    ProcessTemplateType ptd = oW.getProcessTemplate();
    System.out.println("getName= " + ptd.getName());
    System.out.println("getPtid= " + ptd.getPtid());
```

相关任务

第 443 页的『生成代理客户机 (Java Web Service)』

Java Web Service 客户机应用程序使用代理客户机与 Web Service API 进行交互。

第 447 页的『创建 BPEL 流程的辅助控件类 (Java Web Service)』

在诸如 sendMessage 或 call 等具体 API 请求中引用的业务对象要求客户机应用程序使用“document/literal 包装”样式元素。客户机应用程序需要使用辅助控件类来帮助生成必需的包装器元素。

第 445 页的『使用 WSDL2Java 生成代理客户机』

WSDL2Java 是用于生成代理客户机的命令行工具。代理客户机简化了客户机应用程序的编程工作。

添加安全性 (Java Web Service)

必须通过在客户机应用程序中实现安全性机制对 Web Service 通信进行保护。

对于 Web Service API, WebSphere Application Server 当前支持下列安全性机制:

- 用户名令牌
- 轻量级第三方认证 (LTPA)

相关概念

第 56 页的『人员任务的授权角色』

可以对人员任务执行的操作取决于您的授权角色。此角色可以是系统级 J2EE 角色, 也可以是基于实例的角色。

第 28 页的『业务流程的授权角色』

角色是授权级别相同的一组人员。可以对业务流程执行的操作取决于您的授权角色。此角色可以是 J2EE 角色, 也可以是基于实例的角色。

实现用户名令牌

用户名令牌安全性机制提供了用户名和密码凭证。

关于此任务

采用用户名令牌安全性机制时, 您可以选择实现各种回调处理程序。根据选择的不同, 将发生下列其中一种情况:

- 每次运行客户机应用程序时, 都将提示您提供用户名和密码。
- 用户名和密码被写入部署描述符。

在这两种情况下, 提供的用户名和密码都必须与相应业务流程容器或人员任务容器中授权角色的用户名和密码匹配。

用户名和密码将封装在请求消息包络中, 从而以“明文”方式包含在 SOAP 消息头中。因此, 强烈建议将客户机应用程序配置为使用 HTTPS (基于 SSL 的 HTTP) 通信协议。这样, 将对所有通信进行加密。您可以在指定 Web Service API 的端点 URL 地址时选择 HTTPS 通信协议。

要定义用户名令牌, 请执行下列操作:

过程

1. 创建安全性令牌:
 - a. 打开模块的部署编辑器。
 - b. 单击 **WS 扩展** 选项卡。
 - c. 在服务引用中, 可能会列示下列 Web Service 引用:

- service/BFMWSService（对于业务流程）
- service/HTMWSService（对于人员任务）

列示的内容取决于生成代理客户机时是添加了 BFMWS.wsdl（对于业务流程）、HTMWWS.wsdl（对于人员任务）还是两者。

- d. 对于这两种服务引用，请执行下列操作：
 - 1) 选择其中一种服务引用。
 - 2) 展开请求生成者配置部分。
 - 3) 展开安全性令牌部分。
 - 4) 单击添加。将打开安全性令牌窗口。
 - 5) 在名称字段中，输入新安全性令牌的名称：**UserNameTokenBFM** 或 **UserNameTokenHTM**。
 - 6) 在令牌类型下拉列表中，选择用户名。（局部名字段将自动填充缺省值。）
 - 7) 保留 **URI** 字段为空白。用户名令牌不需要 **URI** 值。
 - 8) 单击确定。
2. 创建令牌生成者：
 - a. 打开模块的部署编辑器。
 - b. 单击 **WS** 绑定选项卡。
 - c. 在服务引用中，列示的 Web Service 引用与上一个步骤相同：
 - service/BFMWSService（对于业务流程）
 - service/HTMWSService（对于人员任务）
 - d. 对于这两种服务引用，请执行下列操作：
 - 1) 选择其中一种服务引用。
 - 2) 展开安全性请求生成者绑定配置部分。
 - 3) 展开令牌生成者子部分。
 - 4) 单击添加。将打开“令牌生成者”窗口。
 - 5) 在名称字段中，输入新令牌生成者的名称，例如“UserNameTokenGeneratorBFM”或“UserNameTokenGeneratorHTM”。
 - 6) 在令牌生成者类字段中，确保选中以下令牌生成者类：**com.ibm.wsspi.wssecurity.token.UsernameTokenGenerator**。
 - 7) 在安全性令牌下拉列表中，选择先前创建的相应安全性令牌。
 - 8) 选中使用值类型复选框。
 - 9) 在值类型字段中，选择用户名令牌。（局部名字段将自动填充，以反映您选择的用户名令牌。）
 - 10) 在回调处理程序字段中，输入“com.ibm.wsspi.wssecurity.auth.callback.GUIPromptCallbackHandler”（即运行客户机应用程序时提示输入用户名和密码）或“com.ibm.wsspi.wssecurity.auth.callback.NonPromptCallbackHandler”。
 - 11) 如果选择 **NonPromptCallbackHandler**，那么必须在部署描述符的相应字段中指定有效的用户名和密码。
 - 12) 单击确定。

相关任务

第 436 页的『指定 Web Service 端点地址』

Web Service 端点地址是客户机应用程序访问 Web Service API 时必须指定的 URL。端点地址将被写入为了生成客户机应用程序的代理客户机而导出的 WSDL 文件。

相关信息



IBM WebSphere Developer 技术日志: Web Service 安全性与 WebSphere Application Server V6

实现 LTPA 安全性机制

在先前建立的安全上下文中运行客户机应用程序时，可以使用轻量级第三方认证（LTPA）安全性机制。

关于此任务

仅当客户机应用程序在已建立安全上下文的安全环境中运行时，LTPA 安全性机制才可用。例如，如果客户机应用程序正在 Enterprise JavaBeans (EJB) 容器中运行，那么 EJB 客户机必须先登录，然后才能调用该客户机应用程序。这样，就建立了安全上下文。然后，如果该 EJB 客户机应用程序调用 Web Service，那么 LTPA 回调处理程序将从安全上下文中检索 LTPA 令牌并将其添加到 SOAP 请求消息中。在服务器端，LTPA 机制将对该 LTPA 令牌进行处理。

要实现 LTPA 安全性机制，请执行下列操作：

过程

1. 在 WebSphere Integration Developer 中提供的 Rational Application Developer 环境中，选择 **WS 绑定** → **安全性请求生成者绑定配置** → **令牌生成者**。
2. 创建安全性令牌：
 - a. 打开模块的**部署编辑器**。
 - b. 单击 **WS 扩展** 选项卡。
 - c. 在**服务引用**中，可能会列示下列 **Web Service 引用**：
 - service/BFMWSService（对于业务流程）
 - service/HTMWSService（对于人员任务）

列示的内容取决于生成代理客户机时是添加了 BFMWS.wsdl（对于业务流程）、HTMWWS.wsdl（对于人员任务）还是两者。

- d. 对于这两种服务引用，请执行下列操作：
 - 1) 选择其中一种**服务引用**。
 - 2) 展开**请求生成者配置**部分。
 - 3) 展开**安全性令牌**部分。
 - 4) 单击**添加**。将打开安全性令牌窗口。
 - 5) 在**名称**字段中，输入新安全性令牌的名称：**LTPATokenBFM** 或 **LTPATokenHTM**。
 - 6) 在**令牌类型**下拉列表中，选择 **LTPA 令牌**。（**URI** 和**局部名字段**将自动填充缺省值。）
 - 7) 单击**确定**。
3. 创建令牌生成者：

- a. 打开模块的**部署编辑器**。
- b. 单击 **WS 绑定**选项卡。
- c. 在**服务引用**中，列示的 Web Service 引用与上一个步骤相同：
 - service/BFMWSService（对于业务流程）
 - service/HTMWSService（对于人员任务）
- d. 对于这两种服务引用，请执行下列操作：
 - 1) 选择其中一种**服务引用**。
 - 2) 展开**安全性请求生成者绑定配置**部分。
 - 3) 展开**令牌生成者**子部分。
 - 4) 单击**添加**。将打开“令牌生成者”窗口。
 - 5) 在**名称**字段中，输入新令牌生成者的名称，例如“LTPATokenGeneratorBFM”或“LTPATokenGeneratorHTM”。
 - 6) 在**令牌生成者类**字段中，确保选中以下令牌生成者类：**com.ibm.wsspi.wssecurity.token.LTPATokenGenerator**。
 - 7) 在**安全性令牌**下拉列表中，选择先前创建的相应安全性令牌。
 - 8) 选中**使用值类型**复选框。
 - 9) 在**值类型**字段中，选择 **LTPA** 令牌。（**URI** 和**局部名字**段将自动填充，以反映您选择的 **LTPA** 令牌。）
 - 10) 在**回调处理程序**字段中，输入“com.ibm.wsspi.wssecurity.auth.callback.LTPATokenCallbackHandler”。
 - 11) 单击**确定**。

结果

在运行时，**LTPA** 令牌回调处理程序将从现有安全上下文中检索 LTPA 令牌并将其添加到 SOAP 请求消息中。

添加事务支持（Java Web Service）

可以对 Java Web Service 客户机应用程序进行配置，以便通过在服务请求中传递客户机应用程序上下文来允许服务器端请求处理参与客户机的事务。此原子事务支持是在 Web Services 原子事务（WS-AT）规范中定义的。

关于此任务

WebSphere Application Server 将每个 Web Service API 请求作为独立的原子事务运行。可以将客户机应用程序配置为通过下列其中一种方式使用事务支持：

- **参与事务**。服务器端请求处理将在客户机应用程序的事务上下文中执行。于是，如果服务器在 Web Service API 请求的运行或回滚期间遇到问题，那么客户机应用程序的请求也将回滚。
- **不使用事务支持**。WebSphere Application Server 仍创建新事务来运行该请求，但服务器端请求处理不在客户机应用程序的事务上下文中执行。

在 .NET 环境中开发客户机应用程序

Microsoft .NET 提供了功能强大的开发环境，在此环境中，可以通过 Web Service 来连接应用程序。

生成代理客户机 (.NET)

.NET 客户机应用程序使用代理客户机与 Web Service API 进行交互。代理客户机使客户机应用程序不必处理复杂的 Web Service 消息传递协议。

开始之前

要创建代理客户机，首先必须从 WebSphere 环境中导出多个 WSDL 文件并将它们复制到客户机编程环境。

注：如果您有 WebSphere Process Server 客户机 CD，那么可以从该 CD 复制文件。

关于此任务

代理客户机由一组 C# Bean 类组成。每个类都包含单个 Web Service 所发布的所有方法和对象。服务方法负责将参数组装成完整的 SOAP 消息、通过 HTTP 将 SOAP 消息发送至 Web Service、从 Web Service 接收响应以及处理所返回的数据。

注：代理客户机只需生成一次。一旦生成，所有访问 Web Service API 的客户机应用程序都可以使用同一个代理客户机。

过程

1. 使用 WSDL 命令来生成代理客户机： 输入：

```
wSDL options WSDLfilepath
```

其中：

- *options* 包括：

/language

允许指定用于创建代理类的语言。缺省语言为 C#。您还可以指定 **VB** (Visual Basic)、**JS** (JScript) 或 **VJS** (Visual J#) 作为 *language* 参数。

/output

带有适当后缀的输出文件名。例如，`proxy.cs`。

/protocol

代理类中实现的协议。缺省设置为 **SOAP**。

要获取完整的 **WSDL.exe** 参数列表，请使用 */?* 命令行开关，或者参考 Visual Studio 中 WSDL 工具的联机帮助。

- *WSDLfilepath* 是从 WebSphere 环境导出的或者从客户机 CD 复制的 WSDL 文件的路径和文件名。

以下示例生成人员任务管理器 Web Service API 的代理客户机：

```
wSDL /language:cs /output:proxycient.cs c:\ws\bin\HTMWS.wSDL
```

2. 将该代理客户机编译成动态链接库 (DLL) 文件。

相关任务

第 437 页的『发布 WSDL 文件』

Web 服务定义语言 (WSDL) 文件包含通过 Web Service API 提供的所有操作的详细描述。业务流程管理器 Web Service API 和人员任务管理器 Web Service API 有各自的 WSDL 文件。您首先必须发布这些 WSDL 文件, 然后将它们从 WebSphere 环境复制到开发环境。在开发环境中, 将使用这些 WSDL 文件来生成代理客户机。

创建 BPEL 流程的辅助控件类 (.NET)

某些 Web Service API 操作要求客户机应用程序使用“document/literal”样式包装的元素。客户机应用程序需要使用辅助控件类来帮助生成必需的包装器元素。

开始之前

要创建辅助控件类, 必须先从 WebSphere Process Server 环境中导出 Web Service API 的 WSDL 文件。

关于此任务

Web Service API 的 call() 和 sendMessage() 操作将导致在 WebSphere Process Server 中启动 BPEL 流程。call() 操作的输入消息要求您提供 BPEL 流程输入消息的 document/literal 包装器。要生成 BPEL 流程所需的 Bean 和类, 请将 <wsdl:types> 元素复制到新的 XSD 文件中, 然后使用 xsd.exe 工具生成辅助控件类。

过程

1. 如果尚未从 WebSphere Integration Developer 中导出 BPEL 流程接口的 WSDL 文件, 那么执行此操作。
2. 在文本编辑器或 XML 编辑器中打开该 WSDL 文件。
3. 复制 <wsdl:types> 元素的所有子元素内容并将其粘贴到新的框架 XSD 文件中。
4. 对该 XSD 文件运行 xsd.exe 工具:

```
call xsd.exe file.xsd /classes /o
```

其中:

file.xsd

要转换的 XML 模式定义文件。

/classes (/c)

生成与所指定 XSD 文件的内容相对应的辅助控件类。

/output (/o)

指定所生成的文件的输出目录。如果省略此目录, 那么缺省值是当前目录。

例如:

```
call xsd.exe ProcessCustomer.xsd /classes /output:c:\temp
```

5. 将生成的类文件添加到客户机应用程序中。例如, 如果正在使用 Visual Studio, 那么可以使用项目 → 添加现有项菜单选项完成此操作。

如果 ProcessCustomer.wsdl 文件包含以下内容:

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:bons1="http://com/ibm/bpe/unittest/sca"
  xmlns:tns="http://ProcessTypes/bpel/ProcessCustomer"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="ProcessCustomer"
  targetNamespace="http://ProcessTypes/bpel/ProcessCustomer">
  <wsdl:types>
    <xsd:schema targetNamespace="http://ProcessTypes/bpel/ProcessCustomer"
      xmlns:bons1="http://com/ibm/bpe/unittest/sca"
      xmlns:tns="http://ProcessTypes/bpel/ProcessCustomer"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:import namespace="http://com/ibm/bpe/unittest/sca"
        schemaLocation="xsd-includes/http.com.ibm.bpe.unittest.sca.xsd"/>
      <xsd:element name="doit">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="input1" nillable="true" type="bons1:Customer"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="doitResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="output1" nillable="true" type="bons1:Customer"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="doitRequestMsg">
    <wsdl:part element="tns:doit" name="doitParameters"/>
  </wsdl:message>
  <wsdl:message name="doitResponseMsg">
    <wsdl:part element="tns:doitResponse" name="doitResult"/>
  </wsdl:message>
  <wsdl:portType name="ProcessCustomer">
    <wsdl:operation name="doit">
      <wsdl:input message="tns:doitRequestMsg" name="doitRequest"/>
      <wsdl:output message="tns:doitResponseMsg" name="doitResponse"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>

```

生成的 XSD 文件将包含:

```

<xsd:schema xmlns:bons1="http://com/ibm/bpe/unittest/sca"
  xmlns:tns="http://ProcessTypes/bpel/ProcessCustomer"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://ProcessTypes/bpel/ProcessCustomer">
  <xsd:import namespace="http://com/ibm/bpe/unittest/sca"
    schemaLocation="Customer.xsd"/>
  <xsd:element name="doit">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="input1" type="bons1:Customer" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="doitResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="output1" type="bons1:Customer" nillable="true"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

相关任务

第 439 页的『导出业务对象』

业务流程和人员任务具有明确的接口，通过这些接口，您可以将业务流程和人员任务作为 Web Service 从外部进行访问。如果这些接口引用了业务对象，那么需要将接口定义和业务对象导出到客户机编程环境。

相关信息



Microsoft 的 XML 模式定义工具 (XSD.EXE) 文档

创建客户机应用程序 (.NET)

客户机应用程序将请求发送至 Web Service API 并从 Web Service API 接收响应。通过使用代理客户机来管理通信并使用辅助控件类来格式化复杂的数据类型，客户机应用程序可以调用 Web Service 方法，就象这些方法是局部函数一样。

开始之前

在开始创建客户机应用程序前，请生成代理客户机以及任何所需的辅助控件类。

关于此任务

可以使用 Visual Studio .NET 等与 .NET 兼容的开发工具来开发 .NET 客户机应用程序。可以构建任何类型的 .NET 应用程序，以调用通用 Web Service API。

过程

1. 创建新的客户机应用程序项目。例如，在 Visual Studio 中创建 **WinFX Windows Application**。
2. 在项目选项中，添加对代理客户机的动态链接库 (DLL) 文件的引用。将所有包含业务对象定义的辅助控件类添加到项目中。例如，在 Visual Studio 中，可以使用项目 → 添加现有项选项完成此任务。
3. 创建代理客户机对象。例如：

```
HTMClient.HTMReference.HumanTaskManagerComponent1Export_HumanTaskManagerHttpService service =  
    new HTMClient.HTMReference.HumanTaskManagerComponent1Export_HumanTaskManagerHttpService();
```

4. 声明在要发送到 Web Service 或从 Web Service 接收的消息中使用的业务对象数据类型。例如：

```
HTMClient.HTMReference.TKIID id = new HTMClient.HTMReference.TKIID();
```

```
ClipBG bg = new ClipBG();  
Clip clip = new Clip();
```

5. 调用特定的 Web Service 函数并指定必需参数。例如，要创建并启动人员任务：

```
HTMClient.HTMReference.createAndStartTask task =  
    new HTMClient.HTMReference.createAndStartTask();  
HTMClient.HTMReference.StartTask sTask = new HTMClient.HTMReference.StartTask();
```

```
sTask.taskName = "SimpleTask";  
sTask.taskNamespace = "http://myProcess/com/acme/task";  
sTask.inputMessage = bg;  
task.inputTask = sTask;
```

```
id = service.createAndStartTask(task).outputTask;
```

6. 远程流程和任务由永久 ID 标识，在上一个步骤的示例中，这是 id。例如，要声明先前创建的人员任务：


```
HTMClient.HTMReference.claimTask claim = new HTMClient.HTMReference.claimTask();
claim.inputTask = id;
```

相关任务

第 453 页的『生成代理客户机 (.NET)』

.NET 客户机应用程序使用代理客户机与 Web Service API 进行交互。代理客户机使客户机应用程序不必处理复杂的 Web Service 消息传递协议。

第 454 页的『创建 BPEL 流程的辅助控件类 (.NET)』

某些 Web Service API 操作要求客户机应用程序使用“document/literal”样式包装的元素。客户机应用程序需要使用辅助控件类来帮助生成必需的包装器元素。

添加安全性 (.NET)

通过在客户机应用程序中集成安全性机制，可以对 Web Service 通信进行保护。

关于此任务

这些安全性机制包括用户名令牌（用户名和密码）、二进制定制安全性令牌和基于 XML 的定制安全性令牌。

过程

1. 请下载并安装 Microsoft .NET 的 Web Services Enhancements (WSE) 2.0 SP3。可以从以下网址获得此产品：

<http://www.microsoft.com/downloads/details.aspx?familyid=1ba1f631-c3e7-420a-bc1e-ef18bab66122&displaylang=en>

2. 对生成的代理客户机代码进行修改，如下所示。

将：

```
public class Export1_MyMicroflowHttpService : System.Web.Services.Protocols.SoapHttpClientProtocol {
```

更改为：

```
public class Export1_MyMicroflowHttpService : Microsoft.Web.Services2.WebServicesClientProtocol {
```

注： 如果通过运行 WSDL.exe 工具重新生成代理客户机，这些修改将丢失。

3. 对客户机应用程序代码进行修改，在文件开头添加下列各行：

```
using System.Web.Services.Protocols;
using Microsoft.Web.Services2;
using Microsoft.Web.Services2.Security.Tokens;
...
```

4. 添加代码，以实现所需的安全性机制。例如，下列代码将添加用户名和密码保护：

```
string user = "U1";
string pwd = "password";
UsernameToken token =
    new UsernameToken(user, pwd, PasswordOption.SendPlainText);
me._proxy.RequestSoapContext.Security.Tokens.Clear();
me._proxy.RequestSoapContext.Security.Tokens.Add(token);
```

查询与业务流程和任务相关的对象

可以使用 Web Service API 来查询业务流程编排器数据库中与业务流程和任务相关的对象，以检索这些对象的特定属性。

关于此任务

业务流程编排器数据库存储用于管理业务流程和任务的模板（模型）和实例（运行时）数据。

通过 Web Service API，客户机应用程序可以发出查询，以便从数据库中检索关于业务流程和任务的信息。

客户机应用程序可以发出一次性的查询以检索某个对象的特定属性。您可以将常用查询保存下来。然后，客户机应用程序可以检索并使用这些存储的查询。

对业务流程以及与任务相关的对象执行的查询

使用 Web Service API 的 query 接口来获取有关业务流程和任务的信息。

客户机应用程序使用类似于 SQL 的语法来查询数据库。

Java Web Service 示例

```
string processTemplateName = "ProcessCustomerLR";
query query1 = new query();
query1.selectClause = "DISTINCT PROCESS_INSTANCE.STARTED, PROCESS_INSTANCE.PIID";
query1.whereClause =
    "PROCESS_INSTANCE.TEMPLATE_NAME = '" + processTemplateName + "'";
query1.orderByClause = "PROCESS_INSTANCE.STARTED";
query1.threshold = null;
query1.timeZone = "UTC"; query1.skipTuples = null;
queryResponse queryResponse1 = proxy.query(query1);
```

从数据库检索的信息将作为查询结果集通过 Web Service API 返回。

例如:

```
QueryResultSetType queryResultSet = queryResponse1.queryResultSet;
if (queryResultSet != null) {
    Console.WriteLine("--> QueryResultSetType");
    Console.WriteLine(" . size= " + queryResultSet.size);
    Console.WriteLine(" . numberColumns= " + queryResultSet.numberColumns);
    string indent = " . ";

    // -- the query column info
    QueryColumnInfoType[] queryColumnInfo = queryResultSet.QueryColumnInfo;
    if (queryColumnInfo.Length > 0) {
        Console.WriteLine();
        Console.WriteLine("= . QueryColumnInfoType size= " + queryColumnInfo.Length);
        Console.WriteLine(" | tableName ");
        for (int i = 0; i < queryColumnInfo.Length; i++) {
            Console.WriteLine(" | " + queryColumnInfo[i].tableName.PadLeft(20));
        }
        Console.WriteLine();
        Console.WriteLine(" | columnName ");
        for (int i = 0; i < queryColumnInfo.Length; i++) {
            Console.WriteLine(" | " + queryColumnInfo[i].columnName.PadLeft(20));
        }
        Console.WriteLine();
        Console.WriteLine(" | data type ");
        for (int i = 0; i < queryColumnInfo.Length; i++) {
```

```

        QueryColumnInfoType tt = queryColumnInfo[i].type;
        Console.WriteLine( " | " + tt.ToString());
    }
    Console.WriteLine();
}
else {
    Console.WriteLine("--> queryColumnInfo= <null>");
}

// - the query result values
string[][] result = queryResultSet.result;
if (result !=null) {
    Console.WriteLine();
    Console.WriteLine("= . result size= " + result.Length);
    for (int i = 0; i <& result.Length; i++) {
        Console.Write(indent + i );
        string[] row = result[i];
        for (int j = 0; j <& row.Length; j++ ) {
            Console.Write(" | " + row[j]);
        }
        Console.WriteLine();
    }
}
else {
    Console.WriteLine("--> result= <null>");
}
}
else {
    Console.WriteLine("--> QueryResultSetType= <null>");
}
}

```

此查询函数根据调用者的权限返回对象。查询结果集仅包含调用者有权查看的对象的属性。

可以使用预定义的数据库视图来查询对象属性。对于流程模板而言，查询函数的语法如下所示：

```

ProcessTemplateData[] queryProcessTemplates
    (java.lang.String whereClause,
     java.lang.String orderByClause,
     java.lang.Integer threshold,
     java.util.TimeZone timezone);

```

对于任务模板而言，查询函数的语法如下所示：

```

TaskTemplate[] queryTaskTemplates
    (java.lang.String whereClause,
     java.lang.String orderByClause,
     java.lang.Integer threshold,
     java.util.TimeZone timezone);

```

对于其他业务流程以及与任务相关的对象而言，查询函数的语法如下所示：

```

QueryResultSet query (java.lang.String selectClause,
                     java.lang.String whereClause,
                     java.lang.String orderByClause,
                     java.lang.Integer skipTuples
                     java.lang.Integer threshold,
                     java.util.TimeZone timezone);

```

query 接口还包含 queryAll 方法。可以使用此方法检索某个对象的所有相关数据（例如，以便进行监视）。queryAll 方法的调用者必须具有下列其中一个 Java 2 Platform, Enterprise Edition (J2EE) 角色：

BPESystemAdministrator、BPESystemMonitor、TaskSystemAdministrator 或 TaskSystemMonitor。使用对象的相应工作项进行授权检查在此处不适用。

.NET 示例

```
ProcessTemplateType[] templates = null;

try {
    queryProcessTemplates iW = new queryProcessTemplates();
    iW.whereClause = "PROCESS_TEMPLATE.STATE=PROCESS_TEMPLATE.STATE.STATE_STARTED";
    iW.orderByClause = null;
    iW.threshold = null;
    iW.timeZone = null;

    Console.WriteLine("--> queryProcessTemplates ... ");
    Console.WriteLine("--> query: WHERE " + iW.whereClause + " ORDER BY " +
        iW.orderByClause + " THRESHOLD " + iW.threshold + " TIMEZONE" + iW.timeZone);

    templates = proxy.queryProcessTemplates(iW);

    if (templates.Length < 1) {
        Console.WriteLine("--> No templates found :-(");
    }
    else {
        for (int i = 0; i < templates.Length ; i++) {
            Console.WriteLine("--> found template with ptid: " + templates[i].ptid);
            Console.WriteLine(" and name: " + templates[i].name);
            /* ... other properties of ProcessTemplateType ... */
        }
    }
}
catch (Exception e) {
    Console.WriteLine("exception= " + e);
}
```

查询参数

每个查询都必须指定多个类似于 SQL 的子句和参数。

查询包含以下内容:

- Select 子句
- Where 子句
- Order-by 子句
- Skip-tuples 参数
- Threshold 参数
- Time-zone 参数

相关概念

第 360 页的『Select 子句』

查询函数中的 Select 子句标识查询所要返回的对象属性。

第 360 页的『Where 子句』

查询函数中的 Where 子句描述要对查询范围应用的过滤器条件。

第 362 页的『Order-by 子句』

查询函数中的 Order-By 子句指定查询结果集的排序条件。

第 362 页的『Skip-tuples 参数』

Skip-tuples 参数指定在查询结果集开头要忽略（不返回给该查询结果集中的调用者）的查询结果集元组数。

第 362 页的『Threshold 参数』

查询函数中的 Threshold 参数对通过查询结果集从服务器返回到客户机的对象数进行限制。

第 363 页的『Timezone 参数』

查询函数中的 Timezone 参数定义该查询中的时间戳记常量的时区。

第 364 页的『查询结果』

查询结果集包含查询结果。

对业务流程对象和人员任务对象执行的查询的预定义视图

系统为业务流程对象和人员任务对象提供了预定义的数据库视图。

在查询这些对象的引用数据时，可以使用这些视图。使用这些视图时，不需要对视图列明确添加 JOIN 谓词，系统将自动添加这些构造。可以使用 Web Service API 的查询功能来查询此数据。

相关参考

第 370 页的『ACTIVITY 视图』

这个预定义数据库视图用于查询活动。

第 372 页的『ACTIVITY_ATTRIBUTE 视图』

这个预定义数据库视图用于查询活动的定制属性。

第 372 页的『ACTIVITY_SERVICE 视图』

这个预定义数据库视图用于查询活动服务。

第 372 页的『APPLICATION_COMP 视图』

这个预定义数据库视图用于查询任务的应用程序组件标识和缺省设置。

第 373 页的『ESCALATION 视图』

这个预定义数据库视图用于查询升级数据。

第 374 页的『ESCALATION_CPROP 视图』

这个预定义数据库视图用于查询升级的定制属性。

第 375 页的『ESCALATION_DESC 视图』

这个预定义数据库视图用于查询升级的多语言描述性数据。

第 377 页的『PROCESS_ATTRIBUTE 视图』

这个预定义数据库视图用于查询流程的定制属性。

第 377 页的『PROCESS_INSTANCE 视图』

这个预定义数据库视图用于查询流程实例。

第 378 页的『PROCESS_TEMPLATE 视图』

这个预定义数据库视图用于查询流程模板。

第 378 页的『QUERY_PROPERTY 视图』

这个预定义数据库视图用于查询流程级变量。

第 379 页的『TASK 视图』

这个预定义数据库视图用于查询任务对象。

第 382 页的『TASK_CPROP 视图』

这个预定义数据库视图用于查询任务对象的定制属性。

第 382 页的『TASK_DESC 视图』

这个预定义数据库视图用于查询任务对象的多语言描述性数据。

第 382 页的『TASK_TEMPL 视图』

这个预定义数据库视图包含用于将任务实例化的数据。

第 384 页的『TASK_TEMPL_CPROP 视图』

这个预定义数据库视图用于查询任务模板的定制属性。

第 384 页的『TASK_TEMPL_DESC 视图』

这个预定义数据库视图用于查询任务模板对象的多语言描述性数据。

第 384 页的『WORK_ITEM 视图』

这个预定义数据库视图用于查询流程、任务和升级的工作项和授权数据。

管理存储查询

可以将经常运行的查询保存下来，使它们成为“存储查询”。存储查询可以是可供所有用户使用的查询，即公用查询，也可以是属于特定用户的查询，即专用查询。

关于此任务

存储查询是指存储在数据库中并由名称标识的查询。专用存储查询可以与公用存储查询同名；不同所有者的专用存储查询也可以同名。

您可以使用业务流程对象或任务对象的存储查询，也可以同时使用这两类对象的存储查询。

管理公用存储查询

公用存储查询由系统管理员创建。这些查询可供所有用户使用。

管理其他用户的专用存储查询

专用查询可以由任何用户创建。这些查询只能供该查询的所有者和系统管理员使用。

使用专用存储查询

如果您不是系统管理员，那么可以创建、运行和删除自己的专用存储查询。您还可以使用系统管理员创建的公用存储查询。

第 13 章 开发 JMS 客户机应用程序

您可以开发通过 Java 消息传递服务 (JMS) API 访问业务流程应用程序的客户机应用程序。

关于此任务

JMS 简介

WebSphere Process Server V6.1 支持使用基于 Java 消息传递服务 (JMS) 编程接口的异步消息传递功能作为通信方法。

JMS 为 Java 客户机 (客户机应用程序或 J2EE 应用程序) 提供了以 JMS 消息形式创建、发送、接收和读取请求的通用方法。

JMS 是基于异步消息的接口, 此接口:

- **使用点到点或发布/预订消息传递功能。** 基于消息的框架可以将信息推送到其他应用程序, 而这些应用程序不必明确地发出请求。可以将同一信息同时传递给多个订户。业务流程编排器的 JMS 接口只支持点到点消息传递功能。
- **具有节奏独立性。** JMS 框架以异步方式工作, 但也能够模拟同步请求/响应方式。这使源系统和目标系统能够并行工作, 而不必相互等待。对于业务流程编排器而言, 此功能极为实用, 它使业务流程编排器能够以异步方式与长时间运行业务流程进行交互。
- **支持事务。** 事务使客户机应用程序能够处理作为一个核心单元发送或接收的消息组。JMS 事务在服务器的事务中运行。对于业务流程编排器的 JMS 接口而言, 通常为每个事务发送和接收一条消息。
- **保证信息传递。** JMS 框架能够以事务方式管理消息并确保消息传递 (尽管不保证及时传递)。对于业务流程编排器而言, 由于需要处理业务流程, 因此这种可靠的消息传递能力尤其重要。
- **确保异构框架之间的互操作性。** 源应用程序和目标应用程序可以在异构环境中工作, 而不必处理与各自框架相关的通信和执行问题。
- **使交换更为流畅。** 通过切换到消息方式, 可以交换更详细的信息。

业务流程的要求

使用 WebSphere Integration Developer 开发的对业务流程编排器运行的业务流程必须符合特定的规则, 这样才能通过 JMS API 进行访问。

这些要求包括:

1. 必须使用针对基于 XML 的 RPC 的 Java API (JAX-RPC 1.1) 规范中定义的“document/literal 包装”样式来定义业务流程接口。此样式是使用 WebSphere Integration Developer 开发的所有业务流程和人员任务的缺省样式。
2. 业务流程和人员任务发出的 Web Service 操作故障消息必须包含由 XML 模式元素定义的单一 WSDL 消息部件。例如:

```
<wsdl:part name="myFault" element="myNamespace:myFaultElement"/>
```


相关信息

 针对基于 XML 的 RPC 的 Java API (JAX-RPC) 下载页面

 我应该使用哪种样式的 WSDL?

访问 JMS 接口

要通过 JMS 接口发送和接收消息，应用程序必须先创建与 `BPC.cellname.Bus` 的连接并创建会话，然后生成消息生产者和使用者。

关于此任务

流程服务器接受遵循点到点范例的 Java 消息服务 (JMS) 消息。发送或接收 JMS 消息的应用程序必须执行下列操作。

以下示例假定在受管环境 (EJB、应用程序客户机或 Web 客户机容器) 中执行 JMS 客户机。如果要在 J2SE 环境中执行 JMS 客户机，请参阅“IBM Client for JMS on J2SE with IBM WebSphere Application Server” (<http://www-1.ibm.com/support/docview.wss?uid=swg24012804>)。

过程

1. 创建与 `BPC.cellname.Bus` 的连接。不存在预先配置的适用于客户机应用程序请求的连接工厂：客户机应用程序可以使用 JMS API 的 `ReplyConnectionFactory`，也可以创建自己的连接工厂。在后一种情况下，它可以使用 Java 命名和目录接口 (JNDI) 查询功能来检索连接工厂。JNDI 查询名必须与配置业务流程编排器的外部请求队列时指定的名称相同。以下示例假定客户机应用程序创建它自己的连接工厂“`jms/clientCF`”。

```
//Obtain the default initial JNDI context.  
Context initialContext = new InitialContext();  
  
// Look up the connection factory.  
// Create a connection factory that connects to the BPC bus.  
// Call it, for example, "jms/clientCF".  
// Also configure an appropriate authentication alias.  
ConnectionFactory connectionFactory =  
    (ConnectionFactory)initialContext.lookup("jms/clientCF");
```

```
// Create the connection.  
Connection connection = connectionFactory.createConnection();
```

2. 创建会话，以便可以创建消息生产者和使用者。

```
// Create a transaction session using auto-acknowledgement.  
Session session = connection.createSession(true, Session.AUTO_ACKNOWLEDGE);
```
3. 创建消息生产者以发送消息。JNDI 查询名必须与配置业务流程编排器的外部请求队列时指定的名称相同。

```
// Look up the destination of the Business Process Choreographer input queue to  
// send messages to.  
Queue sendQueue = (Queue) initialContext.lookup("jms/BFMJMSAPIQueue");
```

```
// Create a message producer.  
MessageProducer producer = session.createProducer(sendQueue);
```

4. 创建消息使用者以接收应答。 应答目标的 JNDI 查询名可以指定用户定义的目标, 但也可以指定业务流程编排器定义的缺省应答目标 `.jms/BFMJMSReplyQueue`。在这两种情况下, 应答目标都必须在 `BPC.<cellname>.Bus` 上。

```
// Look up the destination of the reply queue.
Queue replyQueue = (Queue) initialcontext.lookup("jms/BFMJMSReplyQueue");

// Create a message consumer.
MessageConsumer consumer = session.createConsumer(replyQueue);
```

5. 发送消息。

```
// Start the connection.
connection.start();

// Create a message - see the task descriptions for examples - and send it.
// This method is defined elsewhere ...
String payload = createXMLDocumentForRequest();
TextMessage requestMessage = session.createTextMessage(payload);

// Set mandatory JMS header.
// targetFunctionName is the operation name of JMS API
// (for example, getProcessTemplate, sendMessage)
requestMessage.setStringProperty("TargetFunctionName", targetFunctionName);

// Set the reply queue; this is mandatory if the replyQueue
// is not the default queue (as it is in this example).
requestMessage.setJMSReplyTo(replyQueue);

// Send the message.
producer.send(requestMessage);

// Get the message ID.
String jmsMessageID = requestMessage.getJMSMessageID();

session.commit();
```

6. 接收应答。

```
// Receive the reply message and analyse the reply.
TextMessage replyMessage = (TextMessage) consumer.receive();

// Get the payload.
String payload = replyMessage.getText();

session.commit();
```

7. 关闭连接并释放资源。

```
// Final housekeeping; free the resources.
session.close();
connection.close();
```

注: 不必在每个事务完成后关闭连接。一旦建立连接, 可以在关闭该连接前交换任意数目的请求消息和响应消息。以上示例通过单个业务方法中的简单调用来说明一个简单的案例。

业务流程编排器 JMS 消息的结构

每条 JMS 消息的头和主体都必须具有预定义结构。

Java 消息服务 (JMS) 消息包含以下内容:

- 包含消息标识和路由信息的消息头。
- 包含内容的消息体 (有效内容)。

业务流程编排器只支持文本消息格式。

消息头

JMS 允许客户机访问多个消息头字段。

业务流程编排器 JMS 客户机可以设置下列头字段:

- **JMSReplyTo**

对请求作出的应答的发送目标。如果未在请求消息中指定此字段，那么将把应答发送至导出接口的缺省应答目标（导出是业务流程组件的客户机接口显示）。可以使用 `initialContext.lookup("jms/BFMJMSReplyQueue")`；获取此目标。

- **TargetFunctionName**

WSDL 操作的名称，例如“`queryProcessTemplates`”。必须始终设置此字段。注意，`TargetFunctionName` 指定此处描述的通用 JMS 消息接口的操作。请勿将此操作与具体流程或任务提供的可以间接调用（例如使用 `call` 或 `sendMessage` 操作进行调用）的操作混淆。

业务流程编排器客户机还可以访问下列头字段:

- **JMSMessageID**

用于唯一地标识消息。发送消息时，JMS 提供程序将设置此字段。如果客户机在发送消息前设置 `JMSMessageID`，那么它将被 JMS 提供程序覆盖。如果需要使用此消息的标识以便进行认证，那么客户机可以在发送消息后检索 `JMSMessageID`。

- **JMSCorrelationID**

用于链接消息。请不要设置此字段。业务流程编排器应答消息包含请求消息的 `JMSMessageID`。

每条响应消息都包含下列 JMS 头字段:

- **IsBusinessException**

“False”代表 WSDL 输出消息，“true”代表 WSDL 故障消息。

`ServiceRuntimeException` 不会返回到异步客户机应用程序。JMS 请求消息处理期间发生严重异常时，将引起运行时故障，从而导致处理此请求消息的事务回滚。然后，将再次传递该 JMS 请求消息。如果在 SCA 导出处理该消息早期发生故障（例如，对该消息进行反序列化时发生故障），那么将按 SCA 导出接收目标指定的最大失败传递次数进行重试。达到最大失败传递次数后，将把该请求消息添加到业务流程编排器总线的系统异常目标。但是，如果业务流程管理器的 SCA 组件实际处理该请求期间发生故障，那么 WebSphere Process Server 的失败事件管理基础结构将处理失败的请求消息，即，如果重试操作未解决异常情况，那么最终将该消息写入失败事件管理数据库。

消息体

JMS 消息体是包含 XML 文档的字符串，该文档代表操作的 `document/literal` 包装器元素。

以下是有效请求消息体的一个简单示例:

```
<?xml version="1.0" encoding="UTF-8"?>
<_6:queryProcessTemplates xmlns:_6="http://www.ibm.com/xmlns/prod/
websphere/business-process/services/6.0">
<whereClause>PROCESS_TEMPLATE.STATE IN (1)</whereClause>
</_6:queryProcessTemplates>
```

相关任务

第 469 页的『检查响应消息以查找业务异常』

JMS 客户机应用程序必须检查所有响应消息的消息头以查找业务异常。

JMS 授权的呈示

要授权使用 JMS 接口，必须在 WebSphere Application Server 中启用安全性设置。

安装业务流程容器时，角色 **JMSAPIUser** 必须映射到某个用户标识。此用户标识用于发出所有 JMS API 请求。例如，如果 **JMSAPIUser** 映射到“用户 A”，那么所有 JMS API 请求对于流程引擎而言都来自“用户 A”。

必须将下列权限授予 **JMSAPIUser** 角色：

请求	所需的权限
forceTerminate	流程管理员
sendEvent	潜在活动所有者或流程管理员

注： 对于所有其他请求而言，不需要进行特殊的授权。

具有业务流程管理员角色的人员将被授予特权。业务流程管理员是一个特殊角色；它与流程实例的流程管理员不同。业务流程管理员拥有所有特权。

只要流程实例存在，就不能从用户注册表中删除流程启动者的用户标识。如果将此用户标识删除，那么此流程的导航将无法继续进行，并且系统日志文件将包含以下异常：

```
no unique ID for: <user ID>
```

JMS API 概述

JMS 消息接口（以下称为“JMS API”）使您能够开发客户机应用程序，以便以异步方式访问业务流程编排器环境中运行的业务流程。

JMS API 允许客户机应用程序以异步方式与微流程和长时间运行流程进行交互。

除下列例外以外，JMS API 发布的接口与 Web Service API 相同：

- 对于 Web Service API 而言，call 操作只能用于调用微流程。但是，通过使用 JMS API，可以使用 call 操作来调用微流程和长时间运行流程。
- JMS API 未发布下列操作：
 - callAsync 操作及其相关联的回调操作。
 - completeAndClaimSuccessor 和 getParticipatingTask 操作

示例 - 执行长时间运行流程

要让常规客户机应用程序处理长时间运行流程，步骤序列如下所示：

1. 设置 JMS 环境，如第 464 页的『访问 JMS 接口』所述。
2. 获取已安装的流程定义的列表：
 - 发送 `queryProcessTemplates`。
 - 这将返回 **ProcessTemplate** 对象列表。
3. 获取启动活动的列表（使用 `createInstance="yes"` 进行接收或检取）：
 - 发送 `getStartActivities`。
 - 这将返回 **InboundOperationTemplate** 对象列表。
4. 创建输入消息。此操作随环境的不同而有所变化，并可能要求使用预先部署的特定于流程的工件。
5. 创建流程实例：
 - 发出 `sendMessage`。

对于 JMS API 而言，还可以使用 `call` 操作与业务流程提供的长时间运行请求/响应操作进行交互。此操作将把操作结果或故障返回给指定的应答目标，即使在很长一段时间过后亦如此。因此，如果使用 `call` 操作，那么不需要使用 `query` 和 `getOutputMessage` 操作即可获取流程输出或故障消息。

6. （可选）通过重复下列步骤，获取流程实例的输出消息：
 - 发出 `query` 以获取流程实例的完成状态。
 - 发出 `getOutputMessage`。
7. （可选）使用该流程发布的其他操作：
 - 使用 `getWaitingActivities` 或 `getActiveEventHandlers` 来获取 **InboundOperationTemplate** 对象列表。
 - 创建输入消息。
 - 使用 `sendMessage` 来发送消息。
8. （可选）通过使用 `getCustomProperties` 和 `setCustomProperties`，获取和设置对该流程或者所包含的活动定义的定制属性。
9. （可选）停止处理流程实例：
 - 发送 `delete` 和 `terminate` 以停止处理长时间运行流程。

开发 JMS 应用程序

JMS 客户机应用程序必须在 Java 2 Enterprise Edition (J2EE) 环境中以 Java 语言开发。

关于此任务

JMS 客户机应用程序通过 JMS API 交换请求消息和响应消息。要创建请求消息，客户机应用程序将在 JMS `TextMessage` 消息体中填充代表相应操作的 `document/literal` 包装器的 XML 元素。

复制工件

为了帮助创建 JMS 客户机应用程序，可以从 WebSphere 环境复制多个工件。

仅当使用 BOXMLSerializer 来创建 JMS 消息体时，才必须使用这些工件。

可以通过两种方法来获取这些工件：

- 从 WebSphere Process Server 环境中发布并导出这些工件。

对于 WebSphere Process Server 6.1 而言，所有客户机工件都在 `install_root\ProcessChoreographer\client` 目录中。对于 JMS API 而言，这些工件是：

BFMIF.wsdl

BFMIF.xsd

BPCGen.xsd

- 从 WebSphere Process Server 客户机 CD 复制文件。

从服务器环境中发布工件

为了便于开发需要访问 JMS API 的客户机应用程序，可以从 WebSphere 服务器环境中发布许多工件。

关于此任务

对于 WebSphere Process Server 6.1 而言，所有客户机工件都在 `was_home\ProcessChoreographer\client` 目录中。对于 JMS API 而言，这些工件是：

BFMIF.wsdl

BFMIF.xsd

BPCGen.xsd

发布这些工件后，需要将它们复制到客户机编程环境。

从客户机 CD 复制文件

WebSphere Process Server 客户机 CD 提供了访问 JMS API 时所需的文件。

过程

1. 访问客户机 CD 并转到 `ProcessChoreographer\client` 目录。
2. 将所需的文件复制到客户机应用程序开发环境。

对于 WebSphere Process Server 6.1 而言，所有客户机工件都在 `\ProcessChoreographer\client` 目录中。对于 JMS API 而言，这些工件是：

BFMIF.wsdl

BFMIF.xsd

BPCGen.xsd

检查响应消息以查找业务异常

JMS 客户机应用程序必须检查所有响应消息的消息头以查找业务异常。

关于此任务

JMS 客户机应用程序必须首先检查响应消息头中的 **IsBusinessException** 属性。

例如:

```
// receive response message
Message receivedMessage = ((JmsProxy) getToBeInvokedUponObject()).receiveMessage();
String strResponse = ((TextMessage) receivedMessage).getText();

if (receivedMessage.getStringProperty("IsBusinessException") {
    // strResponse is a bussiness fault
    // any api can end w/a processFaultMsg
    // the call api also w/a businessFaultMsg
}
else {
    // strResponse is the output message
}
```

相关概念

第 465 页的『业务流程编排器 JMS 消息的结构』
每条 JMS 消息的头和主体都必须具有预定义结构。

第 14 章 使用 JSF 组件为业务流程和人员任务开发 Web 应用程序

业务流程编排器提供了多个 JavaServer Faces (JSF) 组件。通过扩展和集成这些组件,可以在 Web 应用程序中添加业务流程和人员任务功能。

关于此任务

可以使用 WebSphere Integration Developer 来构建 Web 应用程序。

过程

1. 创建一个动态项目, 然后更改“Web 项目功能部件”属性以包括 Faces 基本组件。

有关创建 Web 项目的更多信息, 请访问 WebSphere Integration Developer 的信息中心。

2. 添加必备的业务流程编排器资源管理器 Java 归档 (JAR 文件)。

将下列文件添加到项目的 WEB-INF/lib 目录中:

- bpcclientcore.jar
- bfmclientmodel.jar
- htmclientmodel.jar
- bpcjsfcomponents.jar

如果正在远程服务器上部署 Web 应用程序, 那么还需添加下列文件。要以远程方式访问业务流程编排器 API, 必需使用这些文件。

- bpe137650.jar
- task137650.jar

在 WebSphere Process Server 中, 所有这些文件都在以下目录中:

- 在 Windows 系统上: *install_root*\ProcessChoreographer\client
 - 在 UNIX、Linux 和 i5/OS 系统上: *install_root*/ProcessChoreographer/client
3. 在 Web 应用程序部署描述符 web.xml 文件中添加所需的 EJB 引用。

```
<ejb-ref id="EjbRef_1">
  <ejb-ref-name>ejb/BusinessProcessHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <home>com.ibm.bpe.api.BusinessFlowManagerHome</home>
  <remote>com.ibm.bpe.api.BusinessFlowManager</remote>
</ejb-ref>
<ejb-ref id="EjbRef_2">
  <ejb-ref-name>ejb/HumanTaskManagerEJB</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <home>com.ibm.task.api.HumanTaskManagerHome</home>
  <remote>com.ibm.task.api.HumanTaskManager</remote>
</ejb-ref>
<ejb-local-ref id="EjbLocalRef_1">
  <ejb-ref-name>ejb/LocalBusinessProcessHome</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <local-home>com.ibm.bpe.api.LocalBusinessFlowManagerHome</local-home>
  <local>com.ibm.bpe.api.LocalBusinessFlowManager</local>
</ejb-local-ref>
```

```

<ejb-local-ref id="EjbLocalRef_2">
  <ejb-ref-name>ejb/LocalHumanTaskManagerEJB</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <local-home>com.ibm.task.api.LocalHumanTaskManagerHome</local-home>
  <local>com.ibm.task.api.LocalHumanTaskManager</local>
</ejb-local-ref>

```

4. 在 JSF 应用程序中添加业务流程编排器资源管理器 JSF 组件。
 - a. 在 JavaServer Pages (JSP) 文件中添加应用程序所需的标记库引用。通常，需要 JSF 和 HTML 标记库以及 JSF 组件所需的标记库。
 - <%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
 - <%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
 - <%@ taglib uri="http://com.ibm.bpe.jsf/taglib" prefix="bpe" %>
 - b. 在 JSP 页面的主体中添加 <f:view> 标记，并在该 <f:view> 标记中添加 <h:form> 标记。
 - c. 在 JSP 文件中添加 JSF 组件。

根据应用程序的不同，在 JSP 文件中添加 List 组件、Details 组件、CommandBar 组件或 Message 组件。对于每个组件，可以添加多个实例。

- d. 在 JSF 配置文件中对受管 Bean 进行配置。

缺省情况下，配置文件是 faces-config.xml 文件。此文件在 Web 应用程序的 WEB-INF 目录中。

根据在 JSP 文件中添加的组件的不同，还需要在 JSF 配置文件中添加对查询和其他包装器对象的引用。为了确保正确地进行错误处理，还需要在 JSF 配置文件中定义错误页面的错误 Bean 和导航目标。

```

<faces-config>
...
<managed-bean>
  <managed-bean-name>BPCErrror</managed-bean-name>
  <managed-bean-class>com.ibm.bpc.clientcore.util.ErrorBeanImpl
  </managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
...
<navigation-rule>
...
<navigation-case>
  <description>
  The general error page.
  </description>
  <from-outcome>error</from-outcome>
  <to-view-id>/Error.jsp</to-view-id>
</navigation-case>
...
</navigation-rule>
</faces-config>

```

在触发了错误页面的错误情况下，将对错误 Bean 设置此异常。

- e. 实现支持 JSF 组件所需的定制代码。
5. 部署应用程序。

如果正在 Network Deployment 环境中部署应用程序，那么请将目标资源的 Java 命名和目录接口（JNDI）名称更改为单元中业务流程管理器 API 和人员任务管理器 API 的位置值。

- 如果业务流程容器被配置在同一受管单元中的另一服务器上，那么名称结构如下所示：

```
cell/nodes/nodename/servers/servername/com/ibm/bpe/api/BusinessManagerHome
cell/nodes/nodename/servers/servername/com/ibm/task/api/HumanTaskManagerHome
```

- 如果业务流程容器被配置在同一单元中的集群中，那么名称结构如下所示：

```
cell/clusters/clustername/com/ibm/bpe/api/BusinessFlowManagerHome
cell/clusters/clustername/com/ibm/task/api/HumanTaskManagerHome
```

将 EJB 引用映射到 JNDI 名称，或者在 `ibm-web-bnd.xmi` 文件中手动添加引用。

下表列示了引用绑定及其缺省映射。

表 47. 从引用绑定到 JNDI 名称的映射

引用绑定	JNDI 名称	注释
<code>ejb/BusinessProcessHome</code>	<code>com/ibm/bpe/api/BusinessFlowManagerHome</code>	远程会话 Bean
<code>ejb/LocalBusinessProcessHome</code>	<code>com/ibm/bpe/api/BusinessFlowManagerHome</code>	本地会话 Bean
<code>ejb/HumanTaskManagerEJB</code>	<code>com/ibm/task/api/HumanTaskManagerHome</code>	远程会话 Bean
<code>ejb/LocalHumanTaskManagerEJB</code>	<code>com/ibm/task/api/HumanTaskManagerHome</code>	本地会话 Bean

结果

部署的 Web 应用程序将包含业务流程编排器资源管理器组件提供的功能。

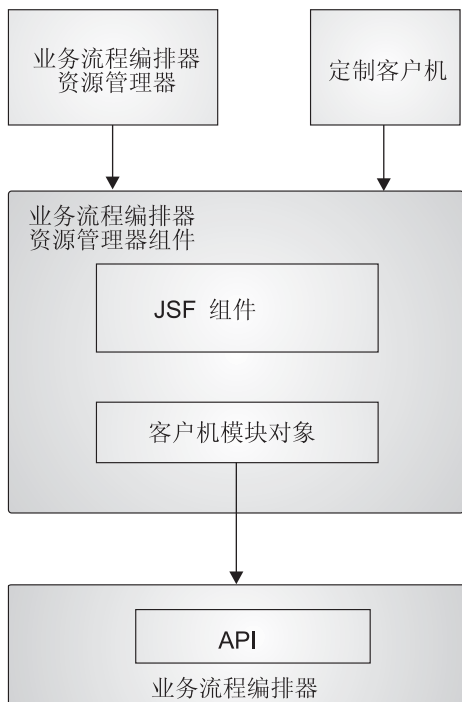
下一步做什么？

如果正在将定制 JSP 用于流程消息和任务消息，那么必须将用于部署 JSP 的 Web 模块映射至定制 JSF 客户机所映射的服务器。

业务流程编排器资源管理器组件

业务流程编排器资源管理器组件是一组可配置的可复用元素，它们基于 JavaServer Faces (JSF) 技术。您可以在 Web 应用程序中嵌入这些元素。这样，这些 Web 应用程序就可以访问已安装的业务流程应用程序和人员任务应用程序。

这些组件由一组 JSF 组件和一组客户机模型对象组成。下图说明这些组件与业务流程编排器、业务流程编排器资源管理器以及其他定制客户机的关系。



JSF 组件

业务流程编排器资源管理器组件包括下列 JSF 组件。在构建要与业务流程和人员任务配合工作的 Web 应用程序时，可以将这些 JSF 组件嵌入到 JavaServer Pages (JSP) 文件中。

- List 组件

List 组件在表中显示应用程序对象列表，这些应用程序对象可以是任务、活动、流程实例、流程模板、工作项或升级。此组件有相关联的列表处理程序。

- Details 组件

Details 组件显示任务、工作项、活动、流程实例和流程模板的属性。此组件有相关联的详细信息处理程序。

- CommandBar 组件

CommandBar 组件显示带有按钮的栏。这些按钮代表要对详细视图中的对象或者列表中所选对象执行的命令。这些对象由列表处理程序或详细信息处理程序提供。

- Message 组件

Message 组件显示可以包含服务数据对象 (SDO) 或简单类型的消息。

客户机模型对象

客户机模型对象与 JSF 组件配合使用。这些对象实现底层业务流程编排器 API 的一些接口并包装原始对象。客户机模型对象提供了对某些属性的标注和转换器的本地语言支持。

JSF 组件中的错误处理

JavaServer Faces (JSF) 组件使用预定义的 `BPCError` 受管 Bean 来处理错误。在触发了错误页面的错误情况下，将对错误 Bean 设置此异常。

这个 Bean 实现了 `com.ibm.bpc.clientcore.util.ErrorBean` 接口。在下列情况下将显示错误页面：

- 在执行对列表处理程序定义的查询期间发生了错误，并且该错误由命令的 `execute` 方法作为 `ClientException` 错误生成
- 如果命令的 `execute` 方法生成了 `ClientException` 错误，并且此错误既不是 `ErrorsInCommandException` 错误也未实现 `CommandBarMessage` 接口
- 如果在该组件中显示了错误消息，并且您转到了该消息的超链接

`com.ibm.bpc.clientcore.util.ErrorBeanImpl` 接口提供了缺省实现。

此接口的定义如下所示：

```
public interface ErrorBean {

    public void setException(Exception ex);

    /*
     * This setter method call allows a locale and
     * the exception to be passed. This allows the
     * getExceptionMessage methods to return localized Strings
     */
    public void setException(Exception ex, Locale locale);

    public Exception getException();
    public String getStack();
    public String getNestedExceptionMessage();
    public String getNestedExceptionStack();
    public String getRootExceptionMessage();
    public String getRootExceptionStack();

    /*
     * This method returns the exception message
     * concatenated recursively with the messages of all
     * the nested exceptions.
     */
    public String getAllExceptionMessages();

    /*
     * This method is returns the exception stack
     * concatenated recursively with the stacks of all
     * the nested exceptions.
     */
    public String getAllExceptionStacks();
}
```

客户机模型对象的缺省转换器和标签

客户机模型对象实现了业务流程编排器 API 的相应接口。

List 组件和 Details 组件可以对任何 Bean 执行操作。您可以显示 Bean 的所有属性。但是，如果要设置用于 Bean 属性的转换器和标签，那么必须使用 List 组件的 `column` 标记或 Details 组件的 `property` 标记。可以通过定义下列静态方法来定义属性的缺省转换器和标签，以代替设置转换器和标签。可以定义下列静态方法：

```

static public String getLabel(String property,Locale locale);
static public com.ibm.bpc.clientcore.converter.SimpleConverter
    getConverter(String property);

```

下表列示了一些客户机模型对象，这些模型对象实现了相应的业务流程管理器 API 类和人员任务管理器 API 类并提供了属性的缺省标签和转换器。接口的这种包装为一组属性提供了与语言环境相关的标注和转换器。下表说明从业务流程编排器接口到相应客户机模型对象的映射。

表 48. 从业务流程编排器接口到客户机模型对象的映射

业务流程编排器接口	客户机模型对象类
com.ibm.bpe.api.ActivityInstanceData	com.ibm.bpe.clientmodel.bean.ActivityInstanceBean
com.ibm.bpe.api.ActivityServiceTemplateData	com.ibm.bpe.clientmodel.bean.ActivityServiceTemplateBean
com.ibm.bpe.api.ProcessInstanceData	com.ibm.bpe.clientmodel.bean.ProcessInstanceBean
com.ibm.bpe.api.ProcessTemplateData	com.ibm.bpe.clientmodel.bean.ProcessTemplateBean
com.ibm.task.api.Escalation	com.ibm.task.clientmodel.bean.EscalationBean
com.ibm.task.api.Task	com.ibm.task.clientmodel.bean.TaskInstanceBean
com.ibm.task.api.TaskTemplate	com.ibm.task.clientmodel.bean.TaskTemplateBean

在 JSF 应用程序中添加 List 组件

使用业务流程编排器资源管理器的 List 组件来显示客户机模型对象（例如业务流程实例或任务实例）的列表。

过程

1. 在 JavaServer Pages (JSP) 文件中添加 List 组件。

在 h:form 标记中添加 bpe:list 标记。bpe:list 标记必须包含 model 属性。在 bpe:list 标记中添加 bpe:column 标记，以添加列表中每一行所要显示的对象属性。

以下示例说明如何添加 List 组件以显示任务实例。

```

<h:form>
    <bpe:list model="#{TaskPool}">
        <bpe:column name="name" action="taskInstanceDetails" />
        <bpe:column name="state" />
        <bpe:column name="kind" />
        <bpe:column name="owner" />
        <bpe:column name="originator" />
    </bpe:list>
</h:form>

```

model 属性引用了名为 TaskPool 的受管 Bean。该受管 Bean 提供了一组 Java 对象，列表将对那些对象执行迭代，然后在各行中显示那些对象。

2. 配置 bpe:list 标记中引用的受管 Bean。

对于 List 组件，这个受管 Bean 必须是 com.ibm.bpe.jsf.handler.BPCListHandler 类的实例。

以下示例说明如何在配置文件中添加 TaskPool 受管 Bean。

```

<managed-bean>
<managed-bean-name>TaskPool</managed-bean-name>
<managed-bean-class>com.ibm.bpe.jsf.handler.BPCListHandler</managed-bean-class>
<managed-bean-scope>session</managed-bean-scope>
  <managed-property>
    <property-name>query</property-name>
    <value>#{TaskPoolQuery}</value>
  </managed-property>
  <managed-property>
    <property-name>type</property-name>
    <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
  </managed-property>
</managed-bean>

<managed-bean>
<managed-bean-name>TaskPoolQuery</managed-bean-name>
<managed-bean-class>sample.TaskPoolQuery</managed-bean-class>
<managed-bean-scope>session</managed-bean-scope>
  <managed-property>
    <property-name>type</property-name>
    <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
  </managed-property>
</managed-bean>

<managed-bean>
<managed-bean-name>htmConnection</managed-bean-name>
<managed-bean-class>com.ibm.task.clientmodel.HTMConnection</managed-bean-class>
<managed-bean-scope>application</managed-bean-scope>
  <managed-property>
    <property-name>jndiName</property-name>
    <value>java:comp/env/ejb/LocalHumanTaskManagerEJB</value>
  </managed-property>
</managed-bean>

```

此示例显示 TaskPool 有两个可配置的属性: query 和 type。query 属性值引用了另一个受管 Bean, 即 TaskPoolQuery。type 属性值指定了 Bean 类, 该类的属性将显示在您所见的列表的列中。相关联的查询实例也可以带有 type 属性。如果指定了 type 属性, 那么它必须与您对列表处理程序指定的 type 相同。

可以对 JSF 应用程序添加任何类型的查询逻辑, 但条件是查询结果可以表示为强类型 Bean 的列表。例如, TaskPoolQuery 是使用一组 com.ibm.task.clientmodel.bean.TaskInstanceBean 对象实现的。

3. 添加列表处理程序所引用的受管 Bean 的定制代码。

以下示例说明如何添加 TaskPool 受管 Bean 的定制代码。

```

public class TaskPoolQuery implements Query {

    public List execute throws ClientException {

        // Examine the faces-config file for a managed bean "htmConnection".
        //
        FacesContext ctx = FacesContext.getCurrentInstance();
        Application app = ctx.getApplication();
        ValueBinding htmVb = app.createValueBinding("#{htmConnection}");
        htmConnection = (HTMConnection) htmVb.getValue(ctx);
        HumanTaskManagerService taskService =
            htmConnection.getHumanTaskManagerService();

        // Then call the actual query method on the Human Task Manager service.
        //
        QueryResultSet queryResult = taskService.query(
            "DISTINCT TASK.TKIID, TASK.NAME, TASK.KIND, TASK.STATE, TASK.TYPE,"
            + "TASK.STARTED, TASK.ACTIVATED, TASK.DUE, TASK.EXPIRES, TASK.PRIORITY" ,

```



```

        "TASK.KIND IN(101,102,105) AND TASK.STATE IN(2)
        AND WORK_ITEM.REASON IN (1)",
        (String)null,
        (Integer)null,
        (TimeZone)null);
    List applicationObjects = transformToTaskList ( queryResult );
    return applicationObjects ;
}

private List transformToTaskList(QueryResultSet result) {

    ArrayList array = null;
    int entries = result.size();
    array = new ArrayList( entries );

    // Transforms each row in the QueryResultSet to a task instance beans.
    for (int i = 0; i < entries; i++) {
        result.next();
        array.add( new TaskInstanceBean( result, connection ));
    }
    return array ;
}
}

```

TaskPoolQuery Bean 查询 Java 对象的属性。此 Bean 必须实现 `com.ibm.bpc.clientcore.Query` 接口。当列表处理程序刷新其内容时，它将调用该查询的 `execute` 方法。此调用将返回 Java 对象列表。`getType` 方法必须返回那些已返回的 Java 对象的类名。

结果

现在，JSF 应用程序包含一个 `JavaServer` 页面，该页面显示所请求的对象列表的属性，例如可供您使用的任务实例的状态、类型、所有者和发起者。

列表的处理方式

List 组件的每个实例都与 `com.ibm.bpe.jsf.handler.BPCListHandler` 类的一个实例相关联。

此列表处理程序记录相关列表中的所选项，并提供了通知机制使列表条目与不同类型的项的详细信息页面相关联。此列表处理程序通过 `bpe:list` 标记的 **model** 属性与 List 组件绑定。

列表处理程序的通知机制通过 `com.ibm.bpe.jsf.handler.ItemListener` 接口实现。您可以在 `JavaServer Faces (JSF)` 应用程序的配置文件中注册此接口的实现。

此通知在列表中的链接被单击时触发。对于所有设置了 **action** 属性的列，都将显示链接。**action** 属性值是 JSF 导航目标或者返回 JSF 导航目标的 JSF 操作方法。

`BPCListHandler` 类还提供了 `refreshList` 方法。您可以在 JSF 方法绑定中使用此方法来实现在于再次运行查询的用户界面控件。

查询实现

可以使用列表处理程序来显示所有类型的对象及其属性。显示的列表内容取决于为列表处理程序配置的 `com.ibm.bpc.clientcore.Query` 接口实现返回的对象列表。可以使用 `BPCListHandler` 类的 `setQuery` 方法通过程序设置查询，也可以在应用程序的 JSF 配置文件中配置查询。

您不仅可以对业务流程编排器 API 运行查询，也可以对应用程序能够访问的任何其他信息来源（例如内容管理系统或数据库）运行查询。唯一的要求是，`execute` 方法必须将查询结果作为对象的 `java.util.List` 返回。

所返回对象的类型必须保证，有相应 `getter` 方法可用于处理定义了该查询的列表列中显示的所有属性。要确保所返回对象的类型适合于列表定义，可以将 Faces 配置文件中定义的 `BPCListHandler` 实例的 `type` 属性值设置为所返回对象的标准类名。可以在查询实现的 `getType` 调用中返回此名称。在运行时，列表处理程序将检查对象类型是否与定义一致。

要将错误消息映射到列表中的特定条目，查询所返回的对象必须实现特征符为 `public Object getID()` 的方法。

缺省转换器和标签

查询返回的项必须是 `Bean`，它们的类必须与作为 `BPCListHandler` 类定义或 `com.ibm.bpc.clientcore.Query` 接口定义中的类型指定的类匹配。此外，`List` 组件将检查项类或超类是否实现了下列方法：

```
static public String getLabel(String property,Locale locale);
static public com.ibm.bpc.clientcore.converter.SimpleConverter
    getConverter(String property);
```

如果对 `Bean` 定义了这些方法，那么 `List` 组件将使用该标签作为列表的缺省标签，并使用 `SimpleConverter` 作为属性的缺省转换器。可以通过 `bpe:list` 标记的 `label` 和 `converterID` 属性覆盖这些设置。有关更多信息，请参阅 `SimpleConverter` 接口和 `ColumnTag` 类的 Javadoc。

特定于用户的时区信息

JavaServer Faces (JSF) 组件提供了实用程序来处理 `List` 组件中特定于用户的时区信息。

`BPCListHandler` 类使用 `com.ibm.bpc.clientcore.util.User` 接口来获取每个用户的时区信息和语言环境信息。`List` 组件期望将此接口的实现中的 `user` 配置成 JavaServer Faces (JSF) 配置文件中的受管 `Bean` 名称。如果配置文件缺少此条目，那么将返回 WebSphere Process Server 使用的时区。

`com.ibm.bpc.clientcore.util.User` 接口的定义如下所示：

```
public interface User {

    /**
     * The locale used by the client of the user.
     * @return Locale.
     */
    public Locale getLocale();

    /**
     * The time zone used by the client of the user.
     * @return TimeZone.
     */
    public TimeZone getTimeZone();

    /**
     * The name of the user.
     * @return name of the user.
     */
    public String getName();
}
```

List 组件中的错误处理

在 JSF 应用程序中使用 List 组件来显示列表时，可以利用 `com.ibm.bpe.jsf.handler.BPCListHandler` 类提供的错误处理功能。

运行查询或执行命令时出错

如果执行查询期间发生错误，那么 `BPCListHandler` 类将对由于访问权不足而导致的错误与其他异常进行区分。要捕获由于访问权不足而导致的错误，查询的 `execute` 方法所抛出的 `ClientException` 的 `rootCause` 参数必须是 `com.ibm.bpe.api.EngineNotAuthorizedException` 或 `com.ibm.task.api.NotAuthorizedException` 异常。List 组件将显示错误消息，而不是显示查询结果。

如果该错误不是由于访问权不足而导致的，那么 `BPCListHandler` 会将该异常对象传递给 JSF 应用程序配置文件中的 `BPCError` 键定义的 `com.ibm.bpc.clientcore.util.ErrorBean` 接口实现。设置该异常时，将调用错误导航目标。

处理列表中显示的项时出错

`BPCListHandler` 类实现了 `com.ibm.bpe.jsf.handler.ErrorHandler` 接口。您可以在 `setErrors` 方法中使用类型为 `java.util.Map` 的映射参数来提供关于这些错误的信息。此映射包含标识（作为键）和异常（作为值）。这些标识必须是引起错误的对象的 `getID` 方法所返回的值。如果设置了该映射，并且任何标识与该列表中显示的任何项相匹配，那么列表处理程序将在该列表中自动添加包含该错误消息的列。

为了避免该列表包含过期的错误消息，请复位错误映射。在下列情况下，该映射将自动复位：

- 调用 `BPCListHandler` 类的 `refreshList` 方法。
- 对 `BPCListHandler` 类设置新查询。
- 使用 `CommandBar` 组件对列表项触发操作。`CommandBar` 组件使用此机制作为其中一种错误处理方法。

List 组件：标记定义

业务流程编排器资源管理器的 List 组件在表中显示对象列表，这些对象可以是任务、活动、流程实例、流程模板、工作项和升级。

List 组件由 JSF 组件标记 `bpe:list` 和 `bpe:column` 组成。`bpe:column` 标记是 `bpe:list` 标记的子元素。

组件类

`com.ibm.bpe.jsf.component.ListComponent`

语法示例

```
<bpe:list model="#{ProcessTemplateList}">
  rows="20"
  styleClass="list"
  headerStyleClass="listHeader"
  rowClasses="normal">

  <bpe:column name="name" action="processTemplateDetails"/>
  <bpe:column name="validFromTime"/>
  <bpe:column name="executionMode" label="Execution mode"/>
```

```

<bpe:column name="state" converterID="my.state.converter"/>
<bpe:column name="autoDelete"/>
<bpe:column name="description"/>

</bpe:list>

```

标记属性

`bpe:list` 标记的主体只能包含 `bpe:column` 标记。显示该表时，List 组件将对应用程序对象列表进行迭代并对每个对象显示所有列。

表 49. `bpe:list` 的属性

属性	必需	描述
<code>buttonStyleClass</code>	否	用于显示页脚区按钮的级联样式表（CSS）样式类。
<code>cellStyleClass</code>	否	用于显示各个表单元格的 CSS 样式类。
<code>checkbox</code>	否	确定是否显示用于选择多个项的复选框。此属性的值为 <code>true</code> 或 <code>false</code> 。如果此值设置为 <code>true</code> ，那么将显示复选框列。
<code>headerStyleClass</code>	否	用于显示表头的 CSS 样式类。
<code>model</code>	是	<code>com.ibm.bpe.jsf.handler.BPCListHandler</code> 类的受管 Bean 的值绑定。
<code>rows</code>	否	每页显示的行数。如果项数超出行数，那么将在表尾显示分页按钮。此属性不支持值表达式。
<code>rowClasses</code>	否	用于显示表行的 CSS 样式类。
<code>selectAll</code>	否	如果此属性设置为 <code>true</code> ，那么缺省情况下将选中列表中所有的项。
<code>styleClass</code>	否	用于显示整个表（包括标题、行和分页按钮）的 CSS 样式类。

表 50. `bpe:column` 的属性

属性	必需	描述
<code>action</code>	否	如果指定此属性，那么将在该列中显示链接。单击此链接时，将触发 JavaServer Faces 操作方法或 Faces 导航目标。JavaServer Faces 操作方法具有以下特征符：String method()。
<code>converterID</code>	否	用于转换属性值的 Faces 转换器的标识。如果未设置此属性，那么将使用模型为此属性提供的任何 Faces 转换器标识。
<code>label</code>	否	这是一个文字或值绑定表达式，用作列头的标签或者表头行单元格的标签。如果未设置此属性，那么将使用模型为此属性提供的任何标签。
<code>name</code>	是	此列中显示的属性的名称。

在 JSF 应用程序中添加 Details 组件

业务流程编排器资源管理器的 Details 组件用于显示任务、工作项、活动、流程实例和流程模板的属性。

过程

1. 在 JavaServer Pages (JSP) 文件中添加 Details 组件。

在 `<h:form>` 标记中添加 `bpe:details` 标记。`bpe:details` 标记必须包含 **model** 属性。可以使用 `bpe:property` 标记对 Details 组件添加属性。

以下示例说明如何添加 Details 组件以显示任务实例的某些属性。

```
<h:form>

    <bpe:details model="#{TaskInstanceDetails}">
        <bpe:property name="displayName" />
        <bpe:property name="owner" />
        <bpe:property name="kind" />
        <bpe:property name="state" />
        <bpe:property name="escalated" />
        <bpe:property name="suspended" />
        <bpe:property name="originator" />
        <bpe:property name="activationTime" />
        <bpe:property name="expirationTime" />
    </bpe:details>

</h:form>
```

model 属性引用了名为 `TaskInstanceDetails` 的受管 Bean。此 Bean 提供了 Java 对象的属性。

2. 配置 `bpe:details` 标记中引用的受管 Bean。

对于 Details 组件，这个受管 Bean 必须是 `com.ibm.bpe.jsf.handler.BPCDetailsHandler` 类的实例。此处理程序类包装 Java 对象并向“详细信息”组件发布其公用属性。

以下示例说明如何在配置文件中添加 `TaskInstanceDetails` 受管 Bean。

```
<managed-bean>
    <managed-bean-name>TaskInstanceDetails</managed-bean-name>
    <managed-bean-class>com.ibm.bpe.jsf.handler.BPCDetailsHandler</managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
    <managed-property>
        <property-name>type</property-name>
        <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
    </managed-property>
</managed-bean>
```

此示例表明 `TaskInstanceDetails` Bean 有一个可配置的 `type` 属性。`type` 属性值指定了 Bean 类 `com.ibm.task.clientmodel.bean.TaskInstanceBean`，这个类的属性将与详细信息显示在同一行中。这个 Bean 类可以是任何 JavaBeans 类。如果这个 Bean 提供了缺省转换器和属性标签，那么该转换器和标签将用于显示，其方式与 List 组件相同。

结果

现在，JSF 应用程序包含一个 JavaServer 页面，该页面将显示所指定的对象的详细信息（例如任务实例的详细信息）。

Details 组件: 标记定义

业务流程编排器资源管理器的 Details 组件用于显示任务、工作项、活动、流程实例和流程模板的属性。

Details 组件由 JSF 组件标记 `bpe:details` 和 `bpe:property` 组成。`bpe:property` 标记是 `bpe:details` 标记的子元素。

组件类

`com.ibm.bpe.jsf.component.DetailsComponent`

语法示例

```
<bpe:details model="#{MyActivityDetails}">
  <bpe:property name="name"/>
  <bpe:property name="owner"/>
  <bpe:property name="activated"/>
</bpe:details>

<bpe:details model="#{MyActivityDetails}" style="style" styleClass="cssStyle">
  style="style"
  styleClass="cssStyle"
</bpe:details>
```

标记属性

使用 `bpe:property` 标记来指定要显示的属性子集以及这些属性的显示顺序。如果 `details` 标记未包含任何属性标记，那么它将显示模型对象的所有可用属性。

表 51. `bpe:details` 的属性

属性	必需	描述
<code>columnClasses</code>	否	用于显示列的一组级联样式表样式（CSS）样式类，以逗号分隔。
<code>id</code>	否	组件的 JavaServer Faces 标识。
<code>model</code>	是	<code>com.ibm.bpe.jsf.handler.BPCDetailsHandler</code> 类的受管 Bean 的值绑定。
<code>rowClasses</code>	否	用于显示行的一组 CSS 样式类，以逗号分隔。
<code>styleClass</code>	否	用于显示 HTML 元素的 CSS 类。

表 52. `bpe:property` 属性

属性	必需	描述
<code>converterID</code>	否	用于在 JavaServer Faces (JSF) 配置文件中注册转换器的标识。
<code>label</code>	否	属性的标注。如果未指定此属性，那么缺省标注由客户机模型类提供。
<code>name</code>	是	要显示的属性的名称。此名称必须与相应客户机模型类中定义的 <code>named</code> 属性相对应。

在 JSF 应用程序中添加 CommandBar 组件

使用业务流程编排器资源管理器的 CommandBar 组件来显示带有按钮的栏。这些按钮代表作用于对象详细视图或列表中的所选对象的命令。

关于此任务

当用户单击用户界面中的按钮时，将对所选对象运行相应的命令。您可以在 JSF 应用程序中添加和扩展 CommandBar 组件。

过程

1. 在 JavaServer Pages (JSP) 文件中添加 CommandBar 组件。

在 `<h:form>` 标记中添加 `bpe:commandbar` 标记。`bpe:commandbar` 标记必须包含 `model` 属性。

以下示例显示如何添加为任务实例列表提供刷新和声明命令的 CommandBar 组件。

```
<h:form>

    <bpe:commandbar model="#{TaskInstanceList}">
        <bpe:command commandID="Refresh" >
            action="#{TaskInstanceList.refreshList}"
            label="Refresh"/>

        <bpe:command commandID="MyClaimCommand" >
            label="Claim" >
            commandClass="<customcode>"/>
        </bpe:commandbar>
    </h:form>
```

model 属性引用了一个受管 Bean。这个 Bean 必须实现 `ItemProvider` 接口并提供选择的 Java 对象。CommandBar 组件通常与同一个 JSP 文件中的 List 组件或 Details 组件配合使用。通常，该标记中指定的 `model` 与同一页面中的 List 组件或 Details 组件中指定的 `model` 相同。因此，对于 List 组件，命令将对列表中的所选项执行操作。

在本示例中，**model** 属性引用了名为 `TaskInstanceList` 的受管 Bean。这个 Bean 在任务实例列表中提供所选对象。这个 Bean 必须实现 `ItemProvider` 接口。此接口由 `BPCListHandler` 类和 `BPCDetailsHandler` 类实现。

2. 可选：配置 `bpe:commandbar` 标记中引用的受管 Bean。

如果 CommandBar 组件的 **model** 属性引用了已配置的受管 Bean（例如列表或详细信息处理程序的受管 Bean），那么不需要进行进一步的配置工作。如果更改任何一个处理程序的配置，或者要使用另一个受管 Bean，那么必须在 JSF 配置文件中添加实现了 `ItemProvider` 接口的受管 Bean。

3. 在 JSF 应用程序中添加实现定制命令的代码。

以下代码段说明如何编写实现 `Command` 接口的命令类。JSP 文件中的 `bpe:command` 标记引用了此命令类 (`MyClaimCommand`)。

```
public class MyClaimCommand implements Command {

    public String execute(List selectedObjects) throws ClientException {
        if( selectedObjects != null && selectedObjects.size() > 0 ) {
```



```

try {
    // Determine HumanTaskManagerService from an HTMConnection bean.
    // Configure the bean in the faces-config.xml for easy access
    // in the JSF application.
    FacesContext ctx = FacesContext.getCurrentInstance();
    ValueBinding vb =
        ctx.getApplication().createValueBinding("{htmConnection}");
    HTMConnection htmConnection = (HTMConnection) htmVB.getValue(ctx);
    HumanTaskManagerService htm =
        htmConnection.getHumanTaskManagerService();

    Iterator iter = selectedObjects.iterator() ;
    while( iter.hasNext() ) {
        try {
            TaskInstanceBean task = (TaskInstanceBean) iter.next() ;
            TKIID tiid = task.getID() ;

            htm.claim( tiid ) ;
            task.setState( new Integer(TaskInstanceBean.STATE_CLAIMED) ) ;

        }
        catch( Exception e ) {
            ; // Error while iterating or claiming task instance.
            // Ignore for better understanding of the sample.
        }
    }
}
catch( Exception e ) {
    ; // Configuration or communication error.
    // Ignore for better understanding of the sample
}
}
return null;
}

// Default implementations
public boolean isMultiSelectEnabled() { return false; }
public boolean[] isApplicable(List itemsOnList) {return null; }
public void setContext(Object targetModel) {; // Not used here }
}

```

命令按如下方式进行处理:

- a. 当用户单击命令栏中的相应按钮时，就调用了命令。CommandBar 组件从 **model** 属性中指定的项提供者检索所选项，并将所选对象列表传递给 **commandClass** 实例的 **execute** 方法。
- b. **commandClass** 属性引用实现了 **Command** 接口的定制命令实现。这意味着该命令必须实现 `public String execute(List selectedObjects) throws ClientException` 方法。此命令返回的结果将用于确定 JSF 应用程序的下一个导航规则。
- c. 此命令完成后，CommandBar 组件将对 **action** 属性进行求值。**action** 属性可以是静态字符串，也可以是具有 `public String Method()` 特征符的 JSF 操作方法的方法绑定。请使用 **action** 属性来覆盖命令类的结果或者明确指定导航规则的结果。如果该命令生成除 `ErrorsInCommandException` 以外的异常，那么不会处理 **action** 属性。
- d. 如果未对 **commandClass** 属性指定命令类，那么将立即调用该操作。例如，对于此示例中的刷新命令，将调用 JSF 值表达式 `#{TaskInstanceList.refreshList}`，而不是调用命令。

结果

现在，JSF 应用程序包含一个 JavaServer 页面，该页面实现了定制的命令栏。

命令的处理方式

可以使用 `CommandBar` 组件在应用程序中添加操作按钮。此组件在用户界面中创建操作按钮并处理单击按钮时创建的事件。

这些按钮触发的功能对 `com.ibm.bpe.jsf.handler.ItemProvider` 接口返回的对象（例如 `BPCListHandler` 类或 `BPCDetailsHandler` 类）执行操作。`CommandBar` 组件使用 `bpe:commandbar` 标记中 **model** 属性值定义的项提供者。

当单击应用程序用户界面上命令栏部分中的按钮时，`CommandBar` 组件将按以下方式处理相关联的事件。

1. `CommandBar` 组件识别对生成该事件的按钮指定的 `com.ibm.bpc.clientcore.Command` 接口实现。
2. 如果与 `CommandBar` 组件相关联的模型实现了 `com.ibm.bpe.jsf.handler.ErrorHandler` 接口，那么将调用 `clearErrorMap` 方法，以便从先前事件中除去错误消息。
3. 调用 `ItemProvider` 接口的 `getSelectedItems` 方法。将返回的项列表传递给该命令的 `execute` 方法并调用该命令。
4. `CommandBar` 组件确定 JavaServer Faces (JSF) 导航目标。如果在 `bpe:commandbar` 标记中未指定 **action** 属性，那么 `execute` 方法的返回值将指定导航目标。如果 **action** 属性设置为 JSF 方法绑定，那么将该方法返回的字符串解释为导航目标。**action** 属性也可以明确指定导航目标。

CommandBar 组件: 标记定义

业务流程编排器资源管理器的 `CommandBar` 组件显示带有按钮的栏。这些按钮作用于详细视图中的对象或列表中的所选对象。

`CommandBar` 组件由 JSF 组件标记 `bpe:commandbar` 和 `bpe:command` 组成。`bpe:command` 标记是 `bpe:commandbar` 标记的子元素。

组件类

`com.ibm.bpe.jsf.component.CommandBarComponent`

语法示例

```
<bpe:commandbar model="#{TaskInstanceList}">
  <bpe:command
    commandID="Work on"
    label="Work on..."
    commandClass="com.ibm.bpc.explorer.command.WorkOnTaskCommand"
    context="#{TaskInstanceDetailsBean}" />
  <bpe:command
    commandID="Cancel"
    label="Cancel"
    commandClass="com.ibm.task.clientmodel.command.CancelClaimTaskCommand"
    context="#{TaskInstanceList}" />
</bpe:commandbar>
```

标记属性

表 53. `bpe:commandbar` 的属性

属性	必需	描述
<code>buttonStyleClass</code>	否	用于显示命令栏按钮的级联样式表（CSS）样式类。
<code>id</code>	否	组件的 JavaServer Faces 标识。
<code>model</code>	是	实现了 <code>ItemProvider</code> 接口的受管 Bean 的值绑定表达式。这个受管 Bean 通常是 <code>CommandBar</code> 组件所在 JavaServer Pages (JSP) 文件中的 <code>List</code> 组件或 <code>Details</code> 组件所使用的 <code>com.ibm.bpe.jsf.handler.BPCListHandler</code> 类或 <code>com.ibm.bpe.jsf.handler.BPCDetailsHandler</code> 类。
<code>styleClass</code>	否	用于显示命令栏的 CSS 样式类。

表 54. `bpe:command` 属性

属性	必需	描述
<code>action</code>	否	要由命令按钮触发的 JavaServer Faces 操作方法或 Faces 导航目标。此操作返回的导航目标将覆盖所有其他导航规则。如果此命令未抛出异常或者抛出 <code>ErrorsInCommandException</code> 异常，那么将调用此操作。
<code>commandClass</code>	否	命令类的名称。类实例由 <code>CommandBar</code> 组件创建，并且在命令按钮被选择时运行。
<code>commandID</code>	是	命令的标识。
<code>context</code>	否	此对象提供 <code>commandClass</code> 属性指定的命令的上下文。第一次访问命令栏时，将检索上下文对象。
<code>immediate</code>	否	指定何时触发命令。如果此属性值为 <code>true</code> ，那么将在处理页面输入前触发命令。缺省值是 <code>false</code> 。
<code>label</code>	是	在命令栏中显示的按钮标注。
<code>rendered</code>	否	确定是否显示按钮。此属性的值可以是布尔值或值表达式。
<code>styleClass</code>	否	用于显示按钮的 CSS 样式类。此样式将覆盖为命令栏定义的按钮样式。

在 JSF 应用程序中添加 Message 组件

使用业务流程编排器资源管理器的 Message 组件在 JavaServer Faces (JSF) 应用程序中显示数据对象和基本类型。

关于此任务

如果消息类型是基本类型，那么将显示标注和输入字段。如果消息类型是数据对象，那么 Message 组件将对该对象进行遍历并显示该对象中的元素。

过程

1. 在 JavaServer Pages (JSP) 文件中添加 Message 组件。

在 `<h:form>` 标记中添加 `bpe:form` 标记。 `bpe:form` 标记必须包含 `model` 属性。

以下示例说明如何添加 Message 组件。

```
<h:form>

    <h:outputText value="Input Message" />
    <bpe:form model="#{MyHandler.inputMessage}" readOnly="true" />

    <h:outputText value="Output Message" />
    <bpe:form model="#{MyHandler.outputMessage}" />

</h:form>
```

Message 组件的 `model` 属性引用了 `com.ibm.bpc.clientcore.MessageWrapper` 对象。此包装器对象包装了服务数据对象 (SDO) 或 Java 基本类型 (例如 `int` 或 `boolean`)。在本示例中, 消息由名为 `MyHandler` 的受管 Bean 的属性提供。

2. 配置 `bpe:form` 标记中引用的受管 Bean。

以下示例说明如何在配置文件中添加 `MyHandler` 受管 Bean。

```
<managed-bean>
<managed-bean-name>MyHandler</managed-bean-name>
<managed-bean-class>com.ibm.bpe.sample.jsf.MyHandler</managed-bean-class>
<managed-bean-scope>session</managed-bean-scope>

    <managed-property>
        <property-name>type</property-name>
        <value>com.ibm.task.clientmodel.bean.TaskInstanceBean</value>
    </managed-property>

</managed-bean>
```

3. 在 JSF 应用程序中添加定制代码。

以下示例说明如何实现输入消息和输出消息。

```
public class MyHandler implements ItemListener {

    private TaskInstanceBean taskBean;
    private MessageWrapper inputMessage, outputMessage

    /* Listener method, e.g. when a task instance was selected in a list handler.
     * Ensure that the handler is registered in the faces-config.xml or manually.
     */
    public void itemChanged(Object item) {
        if( item instanceof TaskInstanceBean ) {
            taskBean = (TaskInstanceBean) item ;
        }
    }

    /* Get the input message wrapper
     */
    public MessageWrapper getInputMessage() {
        try{
            inputMessage = taskBean.getInputMessageWrapper() ;
        }
        catch( Exception e ) {
            ; //...ignore errors for simplicity
        }
        return inputMessage;
    }
}
```

```

/* Get the output message wrapper
*/
public MessageWrapper getOutputMessage() {
    // Retrieve the message from the bean. If there is no message, create
    // one if the task has been claimed by the user. Ensure that only
    // potential owners or owners can manipulate the output message.
    try{
        outputMessage = taskBean.getOutputMessageWrapper();
        if( outputMessage == null
            && taskBean.getState() == TaskInstanceBean.STATE_CLAIMED ) {
            HumanTaskManagerService htm = getHumanTaskManagerService();
            outputMessage = new MessageWrapperImpl();
            outputMessage.setMessage(
                htm.createOutputMessage( taskBean.getID() ).getObject()
            );
        }
    }
    catch( Exception e ) {
        ; //...ignore errors for simplicity
    }
    return outputMessage
}
}
}

```

MyHandler 受管 Bean 实现了 com.ibm.jsf.handler.ItemListener 接口，因此，它可以将其本身注册为列表处理程序的项侦听器。当用户单击列表中的项时，MyHandler Bean 的 itemChanged(Object item) 方法将接收到有关所选项的通知。处理程序将检查项类型，然后存储对相关 TaskInstanceBean 对象的引用。要使用此接口，请在 faces-config.xml 文件中对相应列表处理程序中的 itemListener 列表添加条目。

MyHandler Bean 提供了 getInputMessage 和 getOutputMessage 方法。这两个方法都返回 MessageWrapper 对象。这些方法将调用委托给所引用的任务实例 Bean 进行。例如，如果由于未设置消息而导致任务实例 Bean 返回空，那么该处理程序将创建并存储新的空消息。Message 组件将显示 MyHandler Bean 提供的消息。

结果

现在，JSF 应用程序包含一个 JavaServer 页面，该页面可以显示数据对象和基本类型。

Message 组件：标记定义

业务流程编排器资源管理器的 Message 组件在 JavaServer Faces (JSF) 应用程序中显示 commonj.sdo.DataObject 对象和基本类型（例如整数和字符串）。

Message 组件由 JSF 组件标记 bpe:form 组成。

组件类

com.ibm.bpe.jsf.component.MessageComponent

语法示例

```

<bpe:form model="#{TaskInstanceDetailsBean.inputMessageWrapper}"
    simplification="true" readOnly="true"
    styleClass4table="messageData"
    styleClass4output="messageDataOutput">
</bpe:form>

```

标记属性

表 55. *bpe:form* 的属性

属性	必需	描述
id	否	组件的 JavaServer Faces 标识。
model	是	引用了 <code>commonj.sdo.DataObject</code> 对象或 <code>com.ibm.bpc.clientcore.MessageWrapper</code> 对象的值绑定表达式。
readOnly	否	如果此属性设置为 <code>true</code> ，那么将显示只读表单。缺省情况下，此属性设置为 <code>false</code> 。
simplification	否	如果此属性设置为 <code>true</code> ，那么将显示包含简单类型并且基数为 0 或 1 的属性。缺省情况下，此属性设置为 <code>true</code> 。
style4validinput	否	用于显示有效输入的级联样式表 (CSS) 样式。
style4invalidinput	否	用于显示无效输入的 CSS 样式。
styleClass4invalidInput	否	用于显示无效输入的 CSS 样式类名。
styleClass4output	否	用于显示输出元素的 CSS 样式类名。
styleClass4table	否	用于显示消息组件所显示的表的 CSS 表样式类名。
styleClass4validInput	否	用于显示有效输入的 CSS 样式类名。

第 15 章 开发用于显示任务消息和流程消息的 JSP 页面

业务流程编排器资源管理器界面提供了用于显示和输入业务数据的缺省输入和输出表单。您可以使用 JSP 页面来提供定制的输入和输出表单。

关于此任务

要在 Web 客户端中包括用户定义的 JavaServer Pages (JSP) 页面，您在 WebSphere Integration Developer 中设计人员任务时必须指定这些页面。例如，可以提供用于特定任务及其输入和输出消息的 JSP 页面，也可以提供用于特定用户角色或所有用户角色的 JSP 页面。在运行时，用户定义的 JSP 页面将包括在用于显示输出数据和收集输入数据的用户界面中。

定制的表单不是独立的 Web 页面；它们是业务流程编排器资源管理器在 HTML 表单中嵌入的 HTML 片段，例如消息的所有标签和输入字段的片段。

当用户在包含定制表单的页面中单击按钮时，将提交输入并在业务流程编排器资源管理器中验证该输入。此验证基于所提供的属性类型以及浏览器中使用的语言环境。如果无法对输入进行验证，那么将再次显示同一页面，并且将在 `messageValidationErrors` 请求属性中提供有关验证错误的信息。此信息作为映射提供，此映射将无效属性的 XML 路径表达式 (XPath) 映射至所发生的验证异常。

要对业务流程编排器资源管理器添加定制表单，请使用 WebSphere Integration Developer 完成下列步骤。

过程

1. 创建定制表单。

Web 界面中使用的输入和输出表单的用户定义 JSP 页面需要访问消息数据。请在 JSP 中使用 Java 片段或者使用 JSP 执行语言来访问消息数据。可以通过请求上下文获得表单中的数据。

2. 对任务指定 JSP 页面。

在人员任务编辑器中打开人员任务。在客户机设置中，指定用户定义的 JSP 页面的位置以及该定制表单所适用的角色，例如管理员。业务流程编排器资源管理器的客户机设置存储在任务模板中。在运行时，将随任务模板一起检索这些设置。

3. 将用户定义的 JSP 页面打包到 Web 归档文件 (WAR 文件) 中。

可以将该 WAR 文件与包含任务的模块一起包括在企业归档中，也可以单独部署该 WAR 文件。如果单独部署 JSP，那么请使 JSP 在将要部署业务流程编排器资源管理器或定制客户机的服务器上可用。

如果正在将定制 JSP 用于流程消息和任务消息，那么必须将用于部署 JSP 的 Web 模块映射至定制 JSF 客户机所映射的服务器。

结果

在运行时，定制的表单将显示在业务流程编排器资源管理器中。

用户定义 JSP 片段

用户定义 JavaServer Pages (JSP) 片段嵌入在 HTML 表单标记中。在运行时，业务流程编排器资源管理器会将这些片段包括在所显示的页面中。

输入消息的用户定义 JSP 片段嵌入在输出消息的 JSP 片段前面。

```
<html....>
  ...
  <form...>
    输入 JSP (显示任务输入消息)

    输出 JSP (显示任务输出消息)

  </form>
  ...
</html>
```

由于用户定义 JSP 片段嵌入在 HTML form 标记中，因此，可以添加输入元素。输入元素的名称必须与数据元素的 XML 路径语言 (XPath) 表达式匹配。必须在输入元素名中添加所提供的前缀值作为前缀，这一点十分重要：

```
<input id="address"
  type="text"
  name="{prefix}/selectPromotionalGiftResponse/address"
  value="{messageMap['/selectPromotionalGiftResponse/address']}"
  size="60"
  align="left" />
```

前缀值作为请求属性提供。此属性确保输入名称在外层表单中是唯一的。前缀由业务流程编排器资源管理器生成，您不应对其进行更改：

```
String prefix = (String)request.getAttribute("prefix");
```

仅当可以在给定上下文中对消息进行编辑时，才会设置前缀元素。根据人员任务状态的不同，可以通过不同的方式显示输出数据。例如，如果该任务处于“已声明”状态，那么可以对输出数据进行修改。但是，如果该任务处于“已完成”状态，那么只能显示该数据。在 JSP 片段中，可以测试前缀元素是否存在并相应地显示该消息。以下 JSTL 语句显示了如何测试是否已设置前缀元素。

```
...
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
...
<c:choose>
  <c:when test="{not empty prefix}">
    <!--Read/write mode-->
  </c:when>
  <c:otherwise>
    <!--Read-only mode-->
  </c:otherwise>
</c:choose>
```

第 16 章 创建插件以定制人员任务功能

业务流程编排器提供了事件处理基础结构来处理人员任务处理期间发生的事件。系统还提供了插件点，因此，您可以根据需要对功能进行修改。您可以使用服务供应商接口（SPI）来创建用于处理事件和人员查询的定制插件。

关于此任务

您可以创建用于处理人员任务 API 事件和升级通知事件的插件。还可以创建用于处理人员解析结果的插件。例如，在高峰期，可以对结果列表添加用户以帮助平衡工作负载。

可以在不同的层次注册插件，例如在全局层次对所有任务注册插件、对某个应用程序组件中的任务注册插件、对某个任务模板的所有相关任务注册插件以及对单个任务实例注册插件。

创建 API 事件处理程序

当 API 方法处理人员任务时，将发生 API 事件。请使用 API 事件处理程序插件服务供应商接口（SPI）来创建插件，以处理 API 发送的任务事件或具有相同 API 事件的内部事件。

关于此任务

要创建 API 事件处理程序，请完成下列步骤。

过程

1. 编写实现 `APIEventHandlerPlugin2` 接口或扩展 `APIEventHandler` 实现类的类。此类可以调用其他类的方法。
 - 如果使用 `APIEventHandlerPlugin2` 接口，那么必须实现 `APIEventHandlerPlugin2` 接口和 `APIEventHandlerPlugin` 接口的所有方法。
 - 如果扩展 `SPI` 实现类，那么覆盖所需的方法。

这个类在 Java 2 Enterprise Edition (J2EE) Enterprise JavaBeans (EJB) 应用程序的上下文中运行。请确保这个类及其辅助控件类遵循 EJB 规范。

提示： 如果要从此类中调用 `HumanTaskManagerService` 接口，那么请不要调用将更新生成该事件的任务的方法。此操作将导致数据库死锁。

2. 将该插件类及其辅助控件类组装到 JAR 文件中。

如果辅助控件类由多个 J2EE 应用程序使用，那么可以将这些类打包成一个单独的 JAR 文件并将其注册为共享库。

3. 在 JAR 文件的 `META-INF/services/` 目录中创建插件的服务供应商配置文件。

配置文件提供了用于标识和装入插件的机制。此文件遵循 Java 2 服务供应商接口规范。

- a. 创建名为 `com.ibm.task.spi.plug-in_nameAPIEventHandlerPlugin` 的文件，其中 `plug-in_name` 是插件的名称。

例如，如果插件名为 `Customer`，并且实现了 `com.ibm.task.spi.APIEventHandlerPlugin` 接口，那么配置文件名为 `com.ibm.task.spi.CustomerAPIEventHandlerPlugin`。

- b. 在此文件中的第一个既非注释也非空白的行中，指定步骤 1 中创建的插件类的标准名称。

例如，如果插件类名为 `MyAPIEventHandler`，并且在 `com.customer.plugins` 包中，那么配置文件的第一行必须包含以下条目：
`com.customer.plugins.MyAPIEventHandler`。

结果

您已经有了可安装的 JAR 文件，此文件包含用于处理 API 事件的插件以及可用于装入该插件的服务供应商配置文件。

提示： 只有一个可用于注册 API 事件处理程序和通知事件处理程序的 `eventHandlerName` 属性。如果要同时使用 API 事件处理程序和通知事件处理程序，那么这两个插件实现必须同名，例如，将 `Customer` 用作 SPI 实现的事件处理程序名。

可以使用单个类来实现这两个插件，也可以通过两个不同的类做到这一点。在这两种情况下，都需要在 `META-INF/services/` 目录中创建两个文件，例如 `com.ibm.task.spi.CustomerNotificationEventHandlerPlugin` 和 `com.ibm.task.spi.CustomerAPIEventHandlerPlugin`。

请将插件实现与辅助控件类打包在单个 JAR 文件中。

下一步做什么？

现在，需要安装并注册该插件，以使其在运行时可供人员任务容器使用。可以向任务实例、任务模板或应用程序组件注册 API 事件处理程序。

相关概念

第 49 页的『任务调用方案』
本节描述各种任务调用方法。

API 事件处理程序

当人员任务被修改或更改状态时，将发生 API 事件。要处理这些 API 事件，可以在修改该任务前直接调用事件处理程序（事件前方法），也可以刚好在 API 调用返回前进行调用（事件后方法）。

如果事件前方法抛出 `ApplicationVetoException` 异常，那么不执行 API 操作，将该异常返回给 API 调用者，并且回滚与该事件相关联的事务。如果事件前方法由内部事件触发，并且抛出 `ApplicationVetoException` 异常，那么不执行自动声明等内部事件，但也不会将该异常返回给客户机应用程序。在这种情况下，将在 `SystemOut.log` 文件中写入参考消息。如果 API 方法在处理期间抛出异常，那么将捕获该异常并将其传递至事件后方法。事件后方法返回后，会再次将该异常传递给调用者。

下列规则适用于事件前方法：

- 事件前方法接收相关 API 方法或内部事件的参数。
- 事件前方法可以抛出 `ApplicationVetoException` 异常以防止继续进行处理。

下列规则适用于事件后方法:

- 事件后方法接收提供给 API 调用的参数以及该调用的返回值。如果 API 方法实现抛出异常, 那么事件后方法还将接收该异常。
- 事件后方法不能修改返回值。
- 事件后方法不能抛出异常; 将记录运行时异常, 但将忽略这些异常。

要实现 API 事件处理程序, 可以使用扩展了 `APIEventHandlerPlugin` 接口的 `APIEventHandlerPlugin2` 接口, 也可以扩展缺省的 `com.ibm.task.spi.APIEventHandler` SPI 实现类。如果事件处理程序继承了缺省实现类, 那么将始终实现最新版本的 SPI。如果升级到更高版本的业务流程编排器, 那么可以减少使用新 SPI 方法时必需进行的更改。

由于只能注册一个事件处理程序名, 因此, 如果同时有通知事件处理程序和 API 事件处理程序, 那么这些处理程序必须同名。

创建通知事件处理程序

当人员任务升级时, 将生成通知事件。业务流程编排器提供了用于处理升级的功能, 例如创建升级工作项或发送电子邮件。您可以创建通知事件处理程序, 以便对处理升级的方式进行定制。

关于此任务

要实现通知事件处理程序, 可以使用 `NotificationEventHandlerPlugin` 接口, 也可以扩展缺省的 `com.ibm.task.spi.NotificationEventHandler` 服务供应商接口 (SPI) 实现类。

要创建通知事件处理程序, 请完成下列步骤。

过程

1. 编写实现 `NotificationEventHandlerPlugin` 接口或扩展 `NotificationEventHandler` 实现类的类。此类可以调用其他类的方法。

如果使用 `NotificationEventHandlerPlugin` 接口, 那么必须实现所有接口方法。如果扩展 SPI 实现类, 那么覆盖所需的方法。

这个类在 Java 2 Enterprise Edition (J2EE) Enterprise JavaBeans (EJB) 应用程序的上下文中运行。请确保这个类及其辅助控件类遵循 EJB 规范。

将使用 `EscalationUser` 角色的权限来调用此插件。此角色是人员任务容器配置期间定义的。

提示: 如果要从此类中调用 `HumanTaskManagerService` 接口, 那么不要调用将更新生成该事件的任务或升级的方法。此操作将导致数据库死锁。

2. 将该插件类及其辅助控件类组装到 JAR 文件中。

如果辅助控件类由多个 J2EE 应用程序使用, 那么可以将这些类打包成一个单独的 JAR 文件并将其注册为共享库。

3. 在 JAR 文件的 `META-INF/services/` 目录中创建插件的服务供应商配置文件。

配置文件提供了用于标识和装入插件的机制。此文件遵循 Java 2 服务供应商接口规范。

- a. 创建名为 `com.ibm.task.spi.plugin_nameNotificationEventHandlerPlugin` 的文件，其中 `plug-in_name` 是插件的名称。

例如，如果插件名为 `HelpDeskRequest`（事件处理程序名），并且实现了 `com.ibm.task.spi.NotificationEventHandlerPlugin` 接口，那么配置文件名为 `com.ibm.task.spi.HelpDeskRequestNotificationEventHandlerPlugin`。

- b. 在此文件中的第一个既非注释也非空白的行中，指定步骤 1 中创建的插件类的标准名称。

例如，如果插件类名为 `MyEventHandler`，并且在 `com.customer.plugins` 包中，那么配置文件的第一行必须包含以下条目：
`com.customer.plugins.MyEventHandler`。

结果

您已经有了可安装的 JAR 文件，此文件包含用于处理通知事件的插件以及可用于装入该插件的服务供应商配置文件。可以向任务实例、任务模板或应用程序组件注册 API 事件处理程序。

提示： 只有一个可用于注册 API 事件处理程序和通知事件处理程序的 `eventHandlerName` 属性。如果要同时使用 API 事件处理程序和通知事件处理程序，那么这两个插件实现必须同名，例如，将 `Customer` 用作 SPI 实现的事件处理程序名。

可以使用单个类来实现这两个插件，也可以通过两个不同的类做到这一点。在这两种情况下，都需要在 `META-INF/services/` 目录中创建两个文件，例如 `com.ibm.task.spi.CustomerNotificationEventHandlerPlugin` 和 `com.ibm.task.spi.CustomerAPIEventHandlerPlugin`。

请将插件实现与辅助控件类打包在单个 JAR 文件中。

下一步做什么？

现在，需要安装并注册该插件，以使其在运行时可供人员任务容器使用。可以向任务实例、任务模板或应用程序组件注册通知事件处理程序。

相关概念

第 43 页的『升级』

升级是指人员任务在所指定时间内未被处理时自动生成的警报。例如，如果在定义的时间限制内未声明或未完成任务，那么将生成此警报。对于一个任务，可以指定一个或多个升级。这些升级可以并行启动，也可以作为一连串升级启动。

创建插件以便对人员查询结果进行后处理

人员解析返回对特定角色（例如任务的潜在所有者）指定的用户列表。您可以通过创建插件来更改人员解析所返回的人员查询结果。例如，为了改善工作负载平衡，可以通过创建插件从查询结果中除去工作负载较高的用户。

关于此任务

只能有一个后处理插件；这意味着此插件必须处理所有任务的人员查询结果。此插件可以添加或删除用户，也可以更改用户或组信息。此插件还可以更改结果类型，例如由用户列表更改为组或者更改为“每个人”。

由于此插件在人员解析完成后运行，因此，已经应用了用于维护机密性或安全性的规则。此插件将接收关于人员解析期间除去的用户的信息，此信息将包含在 `HTM_REMOVED_USERS` 映射键中。您必须确保插件使用此上下文信息来维持所有的机密性或安全性规则。

要对人员查询结果进行后处理，请使用 `StaffQueryResultPostProcessorPlugin` 接口。此接口包含用于修改任务查询结果、升级查询结果、任务模板查询结果和应用程序组件查询结果的方法。

要创建插件以便对人员查询结果进行后处理，请完成下列步骤。

过程

1. 编写实现 `StaffQueryResultPostProcessorPlugin` 接口的类。

必须实现所有接口方法。此类可以调用其他类的方法。

这个类在 Java 2 Enterprise Edition (J2EE) Enterprise JavaBeans (EJB) 应用程序的上下文中运行。请确保这个类及其辅助控件类遵循 EJB 规范。

提示： 如果要从此类中调用 `HumanTaskManagerService` 接口，那么请不要调用将重新生成该事件的任务的方法。此操作将导致数据库死锁。

以下示例说明如何更改任务 `SpecialTask` 的编辑者角色。

```
public StaffQueryResult processStaffQueryResult
    (StaffQueryResult originalStaffQueryResult,
     Task task,
     int role,
     Map context)
{
    StaffQueryResult newStaffQueryResult = originalStaffQueryResult;
    StaffQueryResultFactory staffResultFactory =
        StaffQueryResultFactory.newInstance();
    if (role == com.ibm.task.api.WorkItem.REASON_EDITOR &&
        task.getName() != null &&
        task.getName().equals("SpecialTask"))
    {
        UserData user = staffResultFactory.newUserData
            ("SuperEditor",
             new Locale("en-US"),
             "SuperEditor@company.com");
        ArrayList userList = new ArrayList();
        userList.add(user);

        newStaffQueryResult = staffResultFactory.newStaffQueryResult(userList);
    }
    return(newStaffQueryResult);
}
```

2. 将该插件类及其辅助控件类组装到 JAR 文件中。

如果辅助控件类由多个 J2EE 应用程序使用，那么可以将这些类打包成一个单独的 JAR 文件并将其注册为共享库。

3. 在 JAR 文件的 `META-INF/services/` 目录中创建插件的服务供应商配置文件。

配置文件提供了用于标识和装入插件的机制。此文件遵循 Java 2 服务供应商接口规范。

- a. 创建名为 `com.ibm.task.spi.plugin_nameStaffQueryResultPostProcessorPlugin` 的文件，其中 `plugin_name` 是插件的名称。

例如，如果插件名为 `MyHandler`，并且实现了 `com.ibm.task.spi.StaffQueryResultPostProcessorPlugin` 接口，那么配置文件名为 `com.ibm.task.spi.MyHandlerStaffQueryResultPostProcessorPlugin`。

- b. 在此文件中的第一个既非注释也非空白的行中，指定步骤 1 中创建的插件类的标准名称。

例如，如果插件类名为 `StaffPostProcessor`，并且在 `com.customer.plugins` 包中，那么配置文件的第一行必须包含以下条目：

`com.customer.plugins.StaffPostProcessor`。您已经有了可安装的 JAR 文件，此文件包含用于发布流程人员查询结果的插件以及可用于装入该插件的服务供应商配置文件。

4. 安装插件。

只能有一位人员查询结果后处理插件。必须将该插件安装成共享库。

5. 注册插件。

- a. 在管理控制台中，转到人员任务管理器的定制属性页面（应用程序服务器 → **server_name** → 人员任务容器 → 定制属性）。

- b. 添加名为 **Staff.PostProcessorPlugin** 的定制属性并将值设置为插件名，在本示例中使用 `MyHandler`。

相关概念

第 77 页的『共享人员分配』

对于特定的任务角色，任务模板的所有实例使用同一个人员分配条件。这是因为，所有任务实例都根据同一个任务模板进行实例化。为了避免重新运行人员查询，任务模板的各个任务实例共享查询结果。

安装插件

要使用一个插件，必须先安装该插件，以使任务容器能够对其进行访问。

关于此任务

安装插件的方式取决于该插件是仅由一个 Java 2 Enterprise Edition (J2EE) 应用程序使用还是将由多个应用程序使用。

要安装插件，请完成下列其中一个步骤。

- 安装供单个 J2EE 应用程序使用的插件。

将插件的 JAR 文件添加到应用程序 EAR 文件中。在 WebSphere Integration Developer 的部署描述符编辑器中，将该插件的 JAR 文件安装为 Enterprise JavaBeans (EJB) 主模块的 J2EE 应用程序的项目实用程序 JAR 文件。

- 安装供多个 J2EE 应用程序使用的插件。

将该 JAR 文件放入 WebSphere Application Server 共享库并使该库与需要访问该插件的应用程序相关联。要使该 JAR 文件可用于 Network Deployment 环境，请以手动方式将该 JAR 文件分发到每个服务器上，然后对每个单元安装一次共享库。

下一步做什么？

现在，可以注册该插件了。

注册插件

可以在任务容器工件层次结构的不同层次注册插件。例如，在全局层次对所有任务注册插件、对某个应用程序组件的任务注册插件、对某个任务模板的所有相关任务注册插件以及对单个任务实例注册插件。

关于此任务

注册多个插件时，系统支持对作用域进行限定。这意味着在任务容器工件层次结构中较低层次（例如任务实例）注册的插件将替代在较高层次（例如任务模板或应用程序组件）注册的插件。层次结构中的所有层次都支持作用域限定。任务容器将使用在层次结构中最低层次注册的插件。

可以通过下列其中一种方法来注册插件。

- 在任务模型中注册插件。

在 WebSphere Integration Developer 的任务编辑器中，在任务属性区域中“详细信息”页的**事件处理程序名**字段中，指定事件处理程序的名称。

- 为您在运行时创建的临时任务或任务模板注册插件。

使用 TTask 类的 `setEventHandlerName` 方法来注册事件处理程序的名称。

- 在运行时，更改已注册的任务实例事件处理程序。

通过使用 `update(Task task)` 方法，可以在运行时更改任务实例的事件处理程序。调用者必须具有任务管理员权限才能更新此属性。

- 在全局层次注册插件。

在管理控制台中，在人员任务容器的定制属性页中定义插件的定制属性。此定制属性的值是插件名。

第 17 章 安装业务流程应用程序和人员任务应用程序

可以将包含业务流程和/或人员任务的服务组件体系结构（SCA）模块分发到部署目标。部署目标可以是服务器或集群。

开始之前

验证是否已经为每个要安装应用程序的应用程序服务器或集群安装并配置了业务流程管理器和/或人员任务管理器。

关于此任务

例如，可以从管理控制台、从命令行或通过运行管理脚本来安装业务流程应用程序和人员任务应用程序。

结果

安装业务流程或人员任务应用程序后，所有业务流程模板和人员任务模板都将进入“已启动”状态。您可以根据这些模板创建流程实例和任务实例。

下一步做什么？

在可以创建流程实例或任务实例之前，必须启动该应用程序。

相关概念

第 502 页的『业务流程和人员任务的部署』

当 WebSphere Integration Developer 或部署服务生成流程或任务的部署代码时，每个流程组件或任务组件都将映射到一个会话企业 Bean。所有部署代码都将打包到企业应用程序（EAR）文件中。此外，在企业应用程序安装期间，对于每个流程，将生成代表此流程中 Java 代码的 Java 类并将其嵌入到 EAR 文件中。必须将所要部署的每个新版本模型打包到新的企业应用程序中。

『业务流程应用程序和人员任务应用程序在 Network Deployment 环境中的安装方式』

在 Network Deployment 环境中安装流程模板或人员任务模板时，应用程序安装过程将自动执行下列操作。

业务流程应用程序和人员任务应用程序在 Network Deployment 环境中的安装方式

在 Network Deployment 环境中安装流程模板或人员任务模板时，应用程序安装过程将自动执行下列操作。

应用程序以异步方式分阶段安装。每个阶段都必须成功完成，这样才能开始下一个阶段。

1. 开始在 Deployment Manager 上安装该应用程序。

在此阶段，将在 WebSphere 配置库中配置业务流程模板和人员任务模板。并且，还将对该应用程序进行验证。如果发生错误，那么将在 System.out 文件中、在 System.err 文件中或者作为 Deployment Manager 上的 FFDC 条目报告那些错误。

2. 在 Node Agent 上继续安装该应用程序。

在此阶段，将在一个应用程序服务器实例上安装该应用程序。此应用程序服务器实例可以是部署目标的组成部分或者部署目标本身。如果部署目标是包含多个集群成员的集群，那么将从此集群的集群成员中任意选择服务器实例。如果在此阶段发生错误，那么将在 SystemOut.log 文件中、在 SystemErr.log 文件中或者作为 Node Agent 上的 FFDC 条目报告那些错误。

3. 在服务器实例上运行该应用程序。

在此阶段，将把流程模板和人员模板部署到部署目标上的业务流程编排器数据库。如果发生错误，那么将在 System.out 文件中、在 SystemErr.log 文件中或者作为此服务器实例上的 FFDC 条目报告那些错误。

相关任务

第 501 页的第 17 章，『安装业务流程应用程序和人员任务应用程序』

可以将包含业务流程和/或人员任务的服务组件体系结构（SCA）模块分发到部署目标。部署目标可以是服务器或集群。

业务流程和人员任务的部署

当 WebSphere Integration Developer 或部署服务生成流程或任务的部署代码时，每个流程组件或任务组件都将映射到一个会话企业 Bean。所有部署代码都将打包到企业应用程序（EAR）文件中。此外，在企业应用程序安装期间，对于每个流程，将生成代表此流程中 Java 代码的 Java 类并将其嵌入到 EAR 文件中。必须将所要部署的每个新版本模型打包到新的企业应用程序中。

在安装包含业务流程或人员任务的企业应用程序时，这些任务将相应地作为业务流程模板或人员任务模板存储在业务流程编排器数据库中。缺省情况下，新安装的模板处于“已启动”状态。但是，新安装的企业应用程序处于“已停止”状态。您可以逐个启动和停止每个已安装的企业应用程序。

您可以部署许多不同版本的流程模板或任务模板并将各个版本部署在不同的应用程序中。安装新企业应用程序时，将按如下方式确定所要安装的模板版本：

- 如果该模板名和目标名称空间尚未存在，那么将安装新模板
- 如果该模板名和目标名称空间与现有模板的模板名和目标名称空间相同，但生效日期不同，那么将安装现有模板的新版本

注：模板名从组件名派生，而不是从业务流程名或人员任务名派生。

如果未指定生效日期，那么日期按如下方式确定：

- 如果使用 WebSphere Integration Developer，那么生效日期是该人员任务或业务流程的建模日期。
- 如果使用部署服务，那么生效日期是 serviceDeploy 命令的运行日期。只有协作任务将应用程序的安装日期用作生效日期。

相关任务

第 501 页的第 17 章,『安装业务流程应用程序和人员任务应用程序』
可以将包含业务流程和/或人员任务的服务组件体系结构 (SCA) 模块分发到部署目
标。部署目标可以是服务器或集群。

以交互方式安装业务流程应用程序和人员任务应用程序

在运行时,可以使用 `wsadmin` 工具和 `installInteractive` 脚本以交互方式安装应用程序。
可以使用此脚本来更改使用管理控制台安装应用程序时无法更改的设置。

关于此任务

要以交互方式安装业务流程应用程序,请执行下列步骤。

过程

1. 启动 `wsadmin` 工具。

在 `profile_root/bin` 目录中,输入 `wsadmin`。

2. 安装该应用程序。

在 `wsadmin` 命令行提示处,输入以下命令:

```
$AdminApp installInteractive application.ear
```

其中 `application.ear` 是包含流程应用程序的企业归档文件的限定名。将提示您完成
一系列任务,在这些任务中,您可以更改应用程序的值。

3. 保存配置更改。

在 `wsadmin` 命令行提示处,输入以下命令:

```
$AdminConfig save
```

必须保存更改,以便将更新传递至主配置库。如果在未保存更改的情况下结束脚本
编制过程,那么将废弃那些更改。

配置流程应用程序数据源和引用设置

可能需要根据特定数据库基础结构来配置运行了 SQL 语句的流程应用程序。这些 SQL
语句可以来自信息服务活动,也可以是流程安装或实例启动期间运行的语句。

关于此任务

安装应用程序时,可以指定下列类型的数据源:

- 用于在流程安装期间运行 SQL 语句的数据源
- 用于在流程实例启动期间运行 SQL 语句的数据源
- 用于运行 SQL 片段活动的数据源

运行 SQL 片段活动所需的数据源在 `tDataSource` 类型的 BPEL 变量中定义。SQL 片
段活动所需的数据库模式和表名在 `tSetReference` 类型的 BPEL 变量中定义。您可以配
置这两类变量的初始值。

可以使用 `wsadmin` 工具来指定数据源。

过程

1. 使用 `wsadmin` 工具以交互方式安装流程应用程序。
2. 逐步完成各个任务，直到转到用于更新数据源和设置引用的任务为止。

请配置下列环境设置。以下示例说明在这两个任务中可以更改的设置。

3. 保存更改。

示例：使用 `wsadmin` 工具来更新数据源和设置引用

在**更新数据源**任务中，可以更改初始变量值和语句的数据源值，在流程安装期间以及流程启动时将使用这些值。在**更新集合引用**任务中，可以配置与数据库模式和表名相关的设置。

任务 [24]：更新数据源

```
// 更改流程启动时使用的初始变量值的数据源值
```

```
流程名: Test
// 流程模板的名称
流程启动时还是安装流程时: 流程启动时
// 指示是在流程启动时还是在流程安装期间对所指定值进行求值
语句还是变量: 变量
// 指示要更改数据源变量
数据源名: MyDataSource
// 变量的名称
JNDI 名称: [jdbc/sample]:jdbc/newName
// 将 JNDI 名称设置为 jdbc/newName
任务 [25]：更新集合引用
```

```
// 更改用作 BPEL 变量初始值的集合引用值
```

```
流程名: Test
// 流程模板的名称
变量: SetRef
// BPEL 变量名
JNDI 名称: [jdbc/sample]:jdbc/newName
// 将集合引用的数据源 JNDI 名称设置为 jdbc/newName
模式名: [IISAMPLE]
// 数据库模式的名称
模式前缀: []:
// 模式名前缀。
// 仅当生成了模式名时，此设置才适用。
表名: [SETREFTAB]:NEWTABLE
// 将数据库表的名称设置为 NEWTABLE
表前缀: []:
// 表名前缀。
// 仅当生成了表名时，此设置才适用。
```

使用管理控制台卸载业务流程应用程序和人员任务应用程序

可以使用管理控制台来卸载包含业务流程或人员任务的应用程序。

开始之前

要卸载包含业务流程或人员任务的应用程序，必须符合下列先决条件：

- 如果该应用程序安装在独立服务器上，那么该服务器必须处于运行状态并能够访问业务流程编排器数据库。
- 如果该应用程序安装在集群中，那么 `Deployment Manager` 以及至少一个集群成员必须处于运行状态。该集群成员必须能够访问业务流程编排器数据库。

- 如果该应用程序安装在受管服务器上，那么 Deployment Manager 以及此服务器必须处于运行状态。此服务器必须能够访问业务流程编排器数据库。
- 所有属于该应用程序的业务流程模板和人员任务模板都必须处于“已停止”状态。
- 不存在处于任何状态的业务流程模板实例或人员任务模板实例。
-

对于用作开发和单元测试环境的独立服务器环境而言，可以将服务器配置为以开发方式运行。此配置不要求模板已停止并且不存在任何实例。但是，此配置对于生产环境而言无效。

关于此任务

要卸载包含业务流程或人员任务的企业应用程序，请执行下列操作：

过程

1. 停止该应用程序中的所有流程和任务模板。

此操作的目的是防止创建流程和任务实例。

- a. 在管理控制台导航窗格中，单击**应用程序** → **SCA 模块**。
- b. 选择包含所要停止的模板的模块。
- c. 在“其他属性”中，根据情况单击**业务流程**和/或**人员任务**。
- d. 通过单击相应的复选框，选中所有流程和任务模板。
- e. 单击**停止**。

对所有包含业务流程模板或人员任务模板的 EJB 模块重复此步骤。

2. 验证数据库、每个集群的至少一个应用程序服务器以及部署了该应用程序的独立服务器是否处于运行状态。

在 Network Deployment 环境中，对于每个安装了应用程序的集群，Deployment Manager、所有受管独立应用程序服务器以及至少一个应用程序服务器必须处于运行状态。

3. 验证该应用程序是否已没有业务流程实例或人员任务实例。

在必要的时候，管理员可以使用业务流程编排器资源管理器来删除任何流程或任务实例。

4. 停止并卸载该应用程序：

- a. 在管理控制台导航窗格中，单击**应用程序** → **企业应用程序**。
- b. 选择要卸载的应用程序，然后单击**停止**。

如果该应用程序仍包含任何流程实例或任务实例，那么此步骤将失败。

- c. 再次选择要卸载的应用程序，然后单击**卸载**。
- d. 单击**保存**以保存更改。

结果

已卸载该应用程序。

使用管理命令卸载业务流程应用程序和人员任务应用程序

可以使用管理命令来代替管理控制台卸载包含业务流程或人员任务的应用程序。

开始之前

要卸载包含业务流程或人员任务的应用程序，必须符合下列先决条件：

- 如果该应用程序安装在独立服务器上，那么该服务器必须处于运行状态并能够访问业务流程编排器数据库。
- 如果该应用程序安装在集群中，那么 Deployment Manager 以及至少一个集群成员必须处于运行状态。该集群成员必须能够访问业务流程编排器数据库。
- 如果该应用程序安装在受管服务器上，那么 Deployment Manager 以及此服务器必须处于运行状态。此服务器必须能够访问业务流程编排器数据库。
- 所有属于该应用程序的业务流程模板和人员任务模板都必须处于“已停止”状态。
- 不存在处于任何状态的业务流程模板实例或人员任务模板实例。
-

对于用作开发和单元测试环境的独立服务器环境而言，可以将服务器配置为以开发方式运行。此配置不要求模板已停止并且不存在任何实例。但是，此配置对于生产环境而言无效。

此外，如果已启用全局安全性，那么请验证您的用户标识是否具有操作员权限。

确保管理客户机所连接的服务器进程处于运行状态。要确保管理客户机自动连接到该服务器进程，请不要使用 `-conntype NONE` 命令选项。

关于此任务

下列步骤描述如何使用 `bpcTemplates.jacl` 脚本来卸载包含业务流程模板或人员任务模板的应用程序。在可以卸载模板所属的应用程序之前，必须先停止该模板。可以使用 `bpcTemplates.jacl` 脚本来通过一个步骤停止和卸载模板。

在卸载应用程序前，可以删除该应用程序中与模板相关联的流程实例或任务实例，例如使用业务流程编排器资源管理器来删除这些实例。通过在运行 `bpcTemplates.jacl` 脚本时指定 `-force` 选项，也可以在一个步骤中删除任何与模板相关联的实例、停止模板并卸载这些模板。

注意：

由于 `-force` 选项将删除所有流程实例和任务实例数据，因此，使用此选项时务必小心谨慎。

过程

1. 切换到业务流程编排器的样本目录。

在 Windows 平台上，输入以下命令：

```
cd install_root\ProcessChoreographer\admin
```

在 Linux、UNIX 和 i5/OS 平台上，输入以下命令：

```
cd install_root/ProcessChoreographer/admin
```

2. 停止模板并卸载相应的应用程序。

在 Windows 平台上，输入以下命令：

```
install_root\bin\wsadmin -f bpcTemplates.jacl  
    [-user user_name]  
    [-password user password]  
    -uninstall application_name  
    [-force]
```

在 Linux、UNIX 和 i5/OS 平台上，输入以下命令：

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
    [-user user_name]  
    [-password user password]  
    -uninstall application_name  
    [-force]
```

其中：

user_name

如果已启用全局安全性，那么请提供用户标识以便进行认证。

user_password

如果已启用全局安全性，那么请提供用户密码以便进行认证。

application_name

如果已启用全局安全性，那么请提供用户密码以便进行认证。

结果

已卸载该应用程序。

第 5 部分 监视业务流程和任务

第 18 章 监视业务流程和人员任务

开始之前

可以通过 WebSphere Integration Developer 中的“监视”窗格来控制对流程和人员任务执行的监视。无论是否将要启用审计跟踪以及是否要发出事件，都必须采用这种方法。

关于此任务

WebSphere Process Server 包含公共事件基础结构，后者提供了用于管理事件数据的标准格式和机制。

每当发生要求进行监视的情况并且公共事件基础结构服务可用时，业务流程编排器都将发出事件。这些事件遵循公共基本事件规范。您可以使用常规工具来处理这些事件。

也可以使用 Java 片段来创建和发送用户数据事件。有关更多信息，请参阅公共事件基础结构文档中有关发送事件的内容。

第 19 章 监视业务流程事件

为业务流程发出的事件包含与情况无关的数据以及特定于业务流程事件的数据。下面对特定于业务流程事件的属性和元素进行描述。

业务流程事件的事件内容分为下列各类。

业务流程所特有的事件数据

在业务流程中，事件与流程、活动、作用域、链接和变量相关。下面对每一类事件中特定于对象的内容进行描述。

对于业务流程编排器 V6.1 而言，有两种事件格式：

WebSphere Business Monitor 6.0.2 格式

如果存在 WebSphere Integration Developer 6.0.2 中设计的流程，或者在 WebSphere Integration Developer 6.1 中启用了 WebSphere Business Monitor 6.0.2 格式方式，那么将发生 WebSphere Business Monitor 6.0.2 格式的事件。除非另外指定，否则这些事件中特定于对象的内容将作为字符串类型的 XML 元素 *extendedDataElement* 而被写入。

WebSphere Business Monitor 6.1 格式

如果存在 WebSphere Integration Developer 6.1 中设计的流程，并且已启用 WebSphere Business Monitor 6.1 格式方式，那么将发生 WebSphere Business Monitor 6.1 格式的事件。这些事件中特定于对象的内容将作为 XML 元素写入公共基本事件中 *eventPointData* 文件夹中的 *xs:any* 槽，消息有效内容将写入 *applicationData* 部分。XML 的结构在 XML 模式定义 (XSD) 文件 *BFMEvents.xsd* 中定义。此文件在 *install_root\ProcessChoreographer\client* 目录中。

流程

流程实例的事件包含下列特定于对象的事件内容：

属性	描述
<i>processTemplateName</i>	派生实例的流程模板的名称。
<i>processTemplateValidFrom</i>	模板的生效日期。
<i>processTemplateId</i>	流程模板的标识。
<i>processInstanceDescription</i>	可选：流程实例的描述。

属性	描述
processInstanceExecutionState	<p>此字符串值指示流程的状态。其格式为：状态编号-状态描述。此属性可以具有下列其中一个值：</p> <ul style="list-style-type: none"> 1 - STATE_READY 2 - STATE_RUNNING 3 - STATE_FINISHED 4 - STATE_COMPENSATING 5 - STATE_FAILED 6 - STATE_TERMINATED 7 - STATE_COMPENSATED 8 - STATE_TERMINATING 9 - STATE_FAILING 10 - STATE_INDOUBT 11 - STATE_SUSPENDED 12 - STATE_COMPENSATION_FAILED
principal	<p>执行当前步骤时所采用的主体身份。通常，这是流程的启动者。</p>
PayloadType	<p>有效内容类型。此字符串值可以是 none、digest 或 full。此值取决于 WebSphere Integration Developer 中的设置。如果没有可用的显式监视定义，例如，如果选中启用缺省事件，那么缺省有效内容类型为 full。</p>

活动和作用域

活动和作用域包含下列特定于对象的事件内容：

属性	描述
processTemplateName	派生实例的流程模板的名称。
processTemplateValidFrom	模板的生效日期。
activityTemplateName	可选：派生实例的活动模板的名称。
activityInstanceDescription	可选：活动实例的描述。

属性	描述
activityKind	<p>此字符串值标识活动类型。此值的格式为：类型编号-类型描述。此属性可以具有下列其中一个值：</p> <ul style="list-style-type: none"> 3 - KIND_EMPTY 21 - KIND_INVOKE 23 - KIND_RECEIVE 24 - KIND_REPLY 25 - KIND_THROW 26 - KIND_TERMINATE 27 - KIND_WAIT 29 - KIND_COMPENSATE 30 - KIND_SEQUENCE 32 - KIND_SWITCH 34 - KIND_WHILE 36 - KIND_PICK 38 - KIND_FLOW 40 - KIND_SCOPE 42 - KIND_SCRIPT 43 - KIND_STAFF 44 - KIND_ASSIGN 45 - KIND_CUSTOM 46 - KIND_RETHROW 47 - KIND_FOR_EACH_SERIAL 48 - KIND_FOR_EACH_PARALLEL 1000 - SQLSnippet 1001 - RetrieveSet 1002 - InvokeInformationService 1003 - AtomicSQLSnippetSequence
state	<p>此字符串值指示活动的状态。其格式为：状态编号-状态描述。请注意，活动的状态代码与流程的状态代码不同。此属性可以具有下列其中一个值：</p> <ul style="list-style-type: none"> 1 - STATE_INACTIVE 2 - STATE_READY 3 - STATE_RUNNING 4 - STATE_SKIPPED 5 - STATE_FINISHED 6 - STATE_FAILED 7 - STATE_TERMINATED 8 - STATE_CLAIMED 9 - STATE_TERMINATING 10 - STATE_FAILING 11 - STATE_WAITING 12 - STATE_EXPIRED 13 - STATE_STOPPED
bpelId	此字符串值指示活动的 wpc:id 属性。
PayloadType	有效内容类型。此字符串值可以是 none、digest 或 full。此值取决于 WebSphere Integration Developer 中的设置。如果没有可用的显式监视定义，例如，如果选中启用缺省事件，那么缺省有效内容类型为 full。

链接

链接包含下列特定于对象的事件内容:

属性	描述
processTemplateName	派生实例的流程模板的名称。
processTemplateValidFrom	模板的生效日期。
flowBpelId	此字符串值指示包含该链接的流活动的 wpc:id 属性。
elementName	所评估的链接的名称。
description	链接的描述。仅当在流程模型中指定了此属性时，才会包括此属性。
PayloadType	有效内容类型。此字符串值可以是 none、digest 或 full。此值取决于 WebSphere Integration Developer 中的设置。如果没有可用的显式监视定义，例如，如果选中 启用缺省事件 ，那么缺省有效内容类型为 full。

变量

变量包含下列特定于对象的事件内容:

属性	描述
processTemplateName	派生实例的流程模板的名称。
processTemplateValidFrom	模板的生效日期。
variableName	已更改的变量的名称。
variableData	<p>当请求与 WBI 监视器兼容的事件时，将发出此属性。这是变量内容的 XML 表示。每个数据对象属性都以嵌套的扩展数据元素格式报告。元素类型可以是“boolean”或“string”并具有适当的值。如果变量 <i>variableName</i> 尚未初始化，那么没有 <i>variableData</i> 元素。</p> <p>此属性仅适用于 WebSphere Business Monitor 6.0.2 格式的事件。对于 WebSphere Business Monitor 6.1 格式的事件而言，变量内容将写入 <i>applicationData</i> 部分，此部分包含一个名称设置为变量名称的内容元素。</p>
variableData_BO	<p>当请求与 WBI 监视器不兼容的事件时，将发出此属性。此元素的类型为“noValue”，它包含变量内容的 XML 表示。每个数据对象属性都以嵌套的扩展数据元素格式报告。如果变量 <i>variableName</i> 尚未初始化，那么没有 <i>VariableData_BO</i> 元素。</p> <p>此属性仅适用于 WebSphere Business Monitor 6.0.2 格式的事件。对于 WebSphere Business Monitor 6.1 格式的事件而言，变量内容将写入 <i>applicationData</i> 部分，此部分包含一个名称设置为变量名称的内容元素。</p>
bpelId	此字符串值指示活动的 wpc:id 属性。

属性	描述
PayloadType	有效内容类型。此字符串值可以是 none、digest 或 full。此值取决于 WebSphere Integration Developer 中的设置。如果没有可用的显式监视定义，例如，如果选中启用缺省事件，那么缺省有效内容类型为 full。

业务流程事件的扩展名

扩展名指示事件的有效内容。本主题列示业务流程事件的所有扩展名及其相应有效内容。

扩展名包含一个字符串值，此字符串值用作公共基本事件的 *extensionName* 属性值。此字符串值也是用于提供其他事件数据的 XML 元素的名称。事件元素的名称是大写的，例如 BPC.BFM.BASE。XML 元素的名称是混合大小写的，例如 BPCEventCode。除非另有指示，否则所有数据元素的类型都是 string。

业务流程事件可以使用下列扩展名:

- 第 518 页的『 BPC.BFM.BASE 』
- 第 518 页的『 BPC.BFM.PROCESS.BASE 』
- 第 518 页的『 BPC.BFM.PROCESS.STATUS 』
- 第 519 页的『 BPC.BFM.PROCESS.START 』
- 第 519 页的『 BPC.BFM.PROCESS.FAILURE 』
- 第 519 页的『 BPC.BFM.PROCESS.CORREL 』
- 第 519 页的『 BPC.BFM.PROCESS.WISTATUS 』
- 第 519 页的『 BPC.BFM.PROCESS.WITRANSFER 』
- 第 520 页的『 BPC.BFM.PROCESS.ESCALATED 』
- 第 520 页的『 BPC.BFM.PROCESS.EVENT 』
- 第 521 页的『 BPC.BFM.PROCESS.PARTNER 』
- 第 521 页的『 BPC.BFM.ACTIVITY.BASE 』
- 第 522 页的『 BPC.BFM.ACTIVITY.STATUS 』
- 第 522 页的『 BPC.BFM.ACTIVITY.FAILURE 』
- 第 523 页的『 BPC.BFM.ACTIVITY.MESSAGE 』
- 第 523 页的『 BPC.BFM.ACTIVITY.CLAIM 』
- 第 523 页的『 BPC.BFM.ACTIVITY.WISTATUS 』
- 第 523 页的『 BPC.BFM.ACTIVITY.WITRANSFER 』
- 第 524 页的『 BPC.BFM.ACTIVITY.FOREACH 』
- 第 524 页的『 BPC.BFM.ACTIVITY.ESCALATED 』
- 第 524 页的『 BPC.BFM.ACTIVITY.EVENT 』
- 第 524 页的『 BPC.BFM.LINK.STATUS 』
- 第 524 页的『 BPC.BFM.VARIABLE.STATUS 』

BPC.BFM.BASE

BPC.BFM.BASE 继承 WBIMonitoringEvent 的 XML 元素。

表 56. BPC.BFM.BASE 的 XML 元素

XML 元素	描述
<i>BPCEventCode</i>	业务流程编排器事件码，用于标识事件性质。
<i>processTemplateName</i>	流程模板的名称。此名称可以与显示名不同。
<i>processTemplateValidFrom</i>	流程模板的生效时间属性。
<i>eventLocalCounter</i>	局部计数器用于发现同一事务中发生的两个事件的顺序。对于微流程实例而言，此计数器重新构造所发出的所有事件的顺序。对于长时间运行流程而言，局部计数器指示当前导航事务中的顺序。

BPC.BFM.PROCESS.BASE

BPC.BFM.PROCESS.STATUS 继承『BPC.BFM.BASE』的 XML 元素。

表 57. BPC.BFM.PROCESS.BASE 的 XML 元素

XML 元素	描述
<i>processInstanceExecutionState</i>	流程的当前执行状态，格式如下：<状态码>-<状态名>。此属性可以具有下列其中一个值： 1 - STATE_READY 2 - STATE_RUNNING 3 - STATE_FINISHED 4 - STATE_COMPENSATING 5 - STATE_FAILED 6 - STATE_TERMINATED 7 - STATE_COMPENSATED 8 - STATE_TERMINATING 9 - STATE_FAILING 10 - STATE_INDOUBT 11 - STATE_SUSPENDED 12 - STATE_COMPENSATION_FAILED

BPC.BFM.PROCESS.STATUS

BPC.BFM.PROCESS.STATUS 继承『BPC.BFM.PROCESS.BASE』的 XML 元素。

表 58. BPC.BFM.PROCESS.STATUS 的 XML 元素

XML 元素	描述
<i>processTemplateId</i>	流程模板的标识。
<i>processInstanceDescription</i>	流程实例的描述。
<i>principal</i>	与此事件相关联的用户的名称。

BPC.BFM.PROCESS.START

BPC.BFM.PROCESS.START 继承第 518 页的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 59. BPC.BFM.PROCESS.START 的 XML 元素

XML 元素	描述
<i>username</i>	对于 BPC.BFM.PROCESS.START 而言，这是请求启动或重新启动流程的用户的名称。

BPC.BFM.PROCESS.FAILURE

BPC.BFM.PROCESS.FAILURE 继承第 518 页的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 60. BPC.BFM.PROCESS.FAILURE 的 XML 元素

XML 元素	描述
<i>processFailedException</i>	导致流程失败的异常消息。

BPC.BFM.PROCESS.CORREL

BPC.BFM.PROCESS.CORREL 继承第 518 页的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 61. BPC.BFM.PROCESS.CORREL 的 XML 元素

XML 元素	描述
<i>correlationSet</i>	关联集实例，格式如下： <pre><?xml version="1.0"?> <correlationSet name="关联集名"> <property name="属性名" value="属性值"/>* </correlationSet></pre>

BPC.BFM.PROCESS.WISTATUS

BPC.BFM.PROCESS.WISTATUS 继承第 518 页的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 62. BPC.BFM.PROCESS.WISTATUS 的 XML 元素

XML 元素	描述
<i>username</i>	对于 BPC.BFM.PROCESS.WISTATUS 而言，这是已创建或删除工作项的用户的列表。

BPC.BFM.PROCESS.WITRANSFER

BPC.BFM.PROCESS.WITRANSFER 继承第 518 页的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 63. BPC.BFM.PROCESS.WITRANSFER 的 XML 元素

XML 元素	描述
<i>current</i>	工作项的当前所有者的用户名。这是已将工作项移交他人的用户。
<i>target</i>	工作项的新所有者的用户名。

BPC.BFM.PROCESS.ESCALATED

BPC.BFM.PROCESS.ESCALATED 继承第 518 页的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 64. BPC.BFM.PROCESS.ESCALATED 的 XML 元素

XML 元素	描述
<i>escalationName</i>	升级的名称。
<i>operation</i>	对于 BPC.BFM.PROCESS.ESCALATED 而言，这是与所升级直接插入调用任务的事件处理程序相关联的操作。
<i>portTypeName</i>	这是与所升级直接插入调用任务的事件处理程序相关联的操作的端口类型名称。
<i>portTypeNamespace</i>	这是与所升级直接插入调用任务的事件处理程序相关联的操作的端口类型名称空间。

BPC.BFM.PROCESS.EVENT

BPC.BFM.PROCESS.EVENT 继承第 518 页的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 65. BPC.BFM.PROCESS.EVENT 的 XML 元素

XML 元素	描述
<i>message</i> 或 <i>message_BO-</i>	以字符串或业务对象 (BO) 表示的服务输入消息或输出消息。格式取决于是否在 WebSphere Integration Developer 的事件监视器选项卡上选中 监视兼容的事件 选项。 此属性仅适用于 WebSphere Business Monitor 6.0.2 格式的事件。对于 WebSphere Business Monitor 6.1 格式的事件而言，消息内容将写入 <i>applicationData</i> 部分，此部分包含一个名称设置为消息名称的内容元素。
<i>operation</i>	所接收事件的操作名。
<i>portTypeName</i>	这是与事件处理程序相关联的操作的端口类型名称。
<i>portTypeNamespace</i>	这是与事件处理程序相关联的操作的端口类型名称空间。

BPC.BFM.PROCESS.PARTNER

BPC.BFM.PROCESS.PARTNER 继承第 518 页的『BPC.BFM.PROCESS.STATUS』的 XML 元素。

表 66. BPC.BFM.PROCESS.PARTNER 的 XML 元素

XML 元素	描述
<i>partnerLinkName</i>	伙伴链接的名称。

BPC.BFM.ACTIVITY.BASE

BPC.BFM.ACTIVITY.BASE 继承第 518 页的『BPC.BFM.BASE』的 XML 元素。

表 67. BPC.BFM.ACTIVITY.BASE 的 XML 元素

XML 元素	描述
<i>activityKind</i>	活动类型，例如序列活动或调用活动。格式为: <类型码>-<类型名>。此属性可以具有下列其中一个值: 3 - KIND_EMPTY 21 - KIND_INVOKE 23 - KIND_RECEIVE 24 - KIND_REPLY 25 - KIND_THROW 26 - KIND_TERMINATE 27 - KIND_WAIT 29 - KIND_COMPENSATE 30 - KIND_SEQUENCE 32 - KIND_SWITCH 34 - KIND_WHILE 36 - KIND_PICK 38 - KIND_FLOW 40 - KIND_SCOPE 42 - KIND_SCRIPT 43 - KIND_STAFF 44 - KIND_ASSIGN 45 - KIND_CUSTOM 46 - KIND_RETHROW 47 - KIND_FOR_EACH_SERIAL 48 - KIND_FOR_EACH_PARALLEL 1000 - SQLSnippet 1001 - RetrieveSet 1002 - InvokeInformationService 1003 - AtomicSQLSnippetSequence

表 67. BPC.BFM.ACTIVITY.BASE 的 XML 元素 (续)

XML 元素	描述
<i>state</i>	<p>活动实例的当前状态，格式如下：<状态码>-<状态名>。此属性可以具有下列其中一个值：</p> <ul style="list-style-type: none"> 1 - STATE_INACTIVE 2 - STATE_READY 3 - STATE_RUNNING 4 - STATE_SKIPPED 5 - STATE_FINISHED 6 - STATE_FAILED 7 - STATE_TERMINATED 8 - STATE_CLAIMED 9 - STATE_TERMINATING 10 - STATE_FAILING 11 - STATE_WAITING 12 - STATE_EXPIRED 13 - STATE_STOPPED
<i>bpelId</i>	BPEL 文件中活动的 wpc:id 属性。此标识对于流程模型中的所有活动来说是唯一的。

BPC.BFM.ACTIVITY.STATUS

BPC.BFM.ACTIVITY.STATUS 继承第 521 页的『BPC.BFM.ACTIVITY.BASE』的 XML 元素。

表 68. BPC.BFM.ACTIVITY.STATUS 的 XML 元素

XML 元素	描述
<i>activityTemplateName</i>	活动模板的名称。此名称可以与显示名不同。
<i>activityTemplateId</i>	活动模板的内部标识。
<i>activityInstanceDescription</i>	活动实例的描述。
<i>principal</i>	已声明活动的用户的名称。

BPC.BFM.ACTIVITY.FAILURE

BPC.BFM.ACTIVITY.FAILURE 继承『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 69. BPC.BFM.ACTIVITY.FAILURE 的 XML 元素

XML 元素	描述
<i>activityFailedException</i>	导致活动失败的异常。

BPC.BFM.ACTIVITY.MESSAGE

BPC.BFM.ACTIVITY.MESSAGE 继承第 522 页的『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 70. BPC.BFM.ACTIVITY.MESSAGE 的 XML 元素

XML 元素	描述
<i>message</i> 或 <i>message_BO</i>	以字符串或业务对象 (BO) 表示的服务输入消息或输出消息。格式取决于是否在 WebSphere Integration Developer 的事件监视器选项卡上选中监视兼容的事件选项。 此属性仅适用于 WebSphere Business Monitor 6.0.2 格式的事件。对于 WebSphere Business Monitor 6.1 格式的事件而言, 消息内容将写入 <i>applicationData</i> 部分, 此部分包含一个名称设置为消息名称的内容元素。

BPC.BFM.ACTIVITY.CLAIM

BPC.BFM.ACTIVITY.CLAIM 继承第 522 页的『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 71. BPC.BFM.ACTIVITY.CLAIM 的 XML 元素

XML 元素	描述
<i>username</i>	对于 BPC.BFM.ACTIVITY.CLAIM 而言, 这是已声明任务的用户。

BPC.BFM.ACTIVITY.WISTATUS

BPC.BFM.ACTIVITY.WISTATUS 继承第 522 页的『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 72. BPC.BFM.ACTIVITY.WISTATUS 的 XML 元素

XML 元素	描述
<i>username</i>	对于 BPC.BFM.ACTIVITY.WISTATUS 而言, 这是与工作项相关联的用户列表。

BPC.BFM.ACTIVITY.WITRANSFER

BPC.BFM.ACTIVITY.WITRANSFER 继承第 522 页的『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 73. BPC.BFM.ACTIVITY.WITRANSFER 的 XML 元素

XML 元素	描述
<i>current</i>	工作项的当前所有者的用户名。这是已将工作项移交他人的用户。
<i>target</i>	工作项的新所有者的用户名。

BPC.BFM.ACTIVITY.FOREACH

BPC.BFM.ACTIVITY.FOREACH 继承第 522 页的『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 74. BPC.BFM.ACTIVITY.FOREACH 的 XML 元素

XML 元素	描述
<i>parallelBranchesStarted</i>	已启动的分支数。

BPC.BFM.ACTIVITY.ESCALATED

BPC.BFM.ACTIVITY.ESCALATED 继承第 522 页的『BPC.BFM.ACTIVITY.STATUS』的 XML 元素。

表 75. BPC.BFM.ACTIVITY.ESCALATED 的 XML 元素

XML 元素	描述
<i>escalationName</i>	升级的名称。
<i>operation</i>	对于 BPC.BFM.ACTIVITY.ESCALATED 而言，这是与所升级直接插入调用任务的事件处理程序相关联的操作。

BPC.BFM.ACTIVITY.EVENT

BPC.BFM.ACTIVITY.EVENT 继承第 523 页的『BPC.BFM.ACTIVITY.MESSAGE』的 XML 元素。

表 76. BPC.BFM.ACTIVITY.EVENT 的 XML 元素

XML 元素	描述
<i>operation</i>	所接收事件的操作名。

BPC.BFM.LINK.STATUS

BPC.BFM.LINK.STATUS 继承第 518 页的『BPC.BFM.BASE』的 XML 元素。

表 77. BPC.BFM.LINK.STATUS 的 XML 元素

XML 元素	描述
<i>elementName</i>	链接的名称。
<i>description</i>	链接的描述。
<i>flowBpellId</i>	定义了链接的流程活动的标识。

BPC.BFM.VARIABLE.STATUS

BPC.BFM.VARIABLE.STATUS 继承第 518 页的『BPC.BFM.BASE』的 XML 元素。

表 78. BPC.BFM.VARIABLE.STATUS 的 XML 元素

XML 元素	描述
<i>variableName</i>	变量的名称。

表 78. *BPC.BFM.VARIABLE.STATUS* 的 XML 元素 (续)

XML 元素	描述
<i>variableData</i> 或 <i>variableData_BO</i>	<p>如果变量 <i>variableName</i> 尚未初始化, 那么没有 <i>variableData</i> 或 <i>VariableData_BO</i> 元素。此变量的数据以字符串或业务对象 (BO) 表示。格式取决于是否在 WebSphere Integration Developer 的事件监视器选项卡上选中监视兼容的事件选项。</p> <p>此属性仅适用于 WebSphere Business Monitor 6.0.2 格式的事件。对于 WebSphere Business Monitor 6.1 格式的事件而言, 变量内容将写入 <i>applicationData</i> 部分, 此部分包含一个名称设置为变量名称的内容元素。</p>
<i>bpellId</i>	变量的业务流程编排器标识。
<i>principal</i>	更新变量的用户的名称。

业务流程事件

如果您要求对 WebSphere Integration Developer 中的业务流程元素进行监视, 那么将发送业务流程事件。下面列示了业务流程可以发出的所有事件。

当流程状态或活动状态更改时, 将发送事件。业务流程可能会引起下列类型的事件:

- 第 526 页的『流程事件』
- 第 527 页的『活动事件』
- 第 529 页的『活动作用域事件』
- 第 530 页的『链接事件』
- 第 530 页的『变量事件』

XML 模式定义 (XSD) 文件

XML 模式定义 (XSD) 文件 *BFMEvents.xsd* 描述了事件结构。此文件在 *install_root\ProcessChoreographer\client* 目录中。

各个表列的关键点

下列各表中的列包含:

- 代码** 包含事件的编号。对于 WebSphere Business Monitor 6.0.2 格式的事件而言, 此值将作为名为 *BPCEventCode* 的扩展数据元素写入公共基本事件。对于 WebSphere Business Monitor 6.1 格式的事件而言, 此值将写入公共基本事件的 *xs:any* 槽。
- 扩展名** *extensionName* 包含字符串值, 此值定义公共基本事件中包含的特定于事件的信息。此字符串值也是用于提供其他事件数据的 XML 元素的名称。有关扩展名的其他信息, 请参阅第 517 页的『业务流程事件的扩展名』。
- 情况** 引用业务流程事件的情况名。有关情况的详细信息, 请参阅第 530 页的『业务流程事件中的情况』。

事件性质

指向 EventNature 参数中业务流程元素事件情况的指针，这些事件情况将显示在 WebSphere Integration Developer 中。

流程事件

下表对所有流程事件进行描述。

代码	扩展名	情况	事件性质	描述
21000	BPC.BFM.PROCESS.START	启动	ENTRY	流程已启动
21001	BPC.BFM.PROCESS.STATUS	报告	SUSPENDED	流程已暂挂。要暂挂流程实例，请使用业务流程编排器资源管理器。
21002	BPC.BFM.PROCESS.STATUS	报告	RESUMED	流程已恢复。只能恢复已暂挂的流程。要恢复流程实例，请使用业务流程编排器资源管理器。
21004	BPC.BFM.PROCESS.STATUS	停止	EXIT	流程已完成
21005	BPC.BFM.PROCESS.STATUS	停止	TERMINATED	流程已终止。要终止流程实例，请使用业务流程编排器资源管理器。
21019	BPC.BFM.PROCESS.START	报告	RESTARTED	流程已重新启动
21020	BPC.BFM.PROCESS.STATUS	破坏	DELETED	流程已删除
42001	BPC.BFM.PROCESS.FAILURE	失败	FAILED	流程已失败
42003	BPC.BFM.PROCESS.STATUS	报告	COMPENSATING	流程正在进行补偿。要对流程实例进行补偿，请使用业务流程编排器资源管理器。
42004	BPC.BFM.PROCESS.STATUS	停止	COMPENSATED	流程已进行补偿
42009	BPC.BFM.PROCESS.STATUS	报告	TERMINATING	流程正在终止
42010	BPC.BFM.PROCESS.STATUS	报告	FAILING	流程即将失败
42027	BPC.BFM.PROCESS.CORREL	报告	CORRELATION	关联集已初始化。在初始化流程实例的新关联集时，将发出此事件。例如，当带有始发关联集接收活动接收到消息时，情况即如此。
42041	BPC.BFM.PROCESS.WISTATUS	报告	WI_DELETED	流程工作项已删除
42042	BPC.BFM.PROCESS.WISTATUS	报告	WI_CREATED	流程工作项已创建
42046	BPC.BFM.PROCESS.STATUS	失败	COMPFAILED	流程补偿失败

代码	扩展名	情况	事件性质	描述
42047	BPC.BFM.PROCESS.EVENT	报告	EV_RECEIVED	流程事件已接收。要定义事件，请使用流程接口。当激活与流程相关联的事件处理程序时，将生成此事件。
42049	BPC.BFM.PROCESS.ESCALATED	报告	EV_ESCALATED	流程事件已升级。如果流程级定义的与 onEvent 事件处理程序相关联的直接插入调用任务升级时，将生成此事件。
42056	BPC.BFM.PROCESS.WITRANSFER	报告	WI_TRANSFERRED	流程工作项已移交
42058	BPC.BFM.PROCESS.PARTNER	报告	PA_CHANGE	流程伙伴已更改。在对伙伴链接指定新的端点引用时，将生成此事件。

对于流程事件而言，下列事件相关范围标识包含下列内容：

- ECSCurrentID 提供流程实例的标识。
- ECSParentID 提供当前流程的流程实例启动事件发生前的 ECSCurrentID 值。

活动事件

下表对所有活动事件进行描述。

代码	扩展名	情况	事件性质	描述
21006	BPC.BFM.ACTIVITY.MESSAGE	启动	CREATED	活动就绪。当人员任务活动启动时，将生成此事件。
21007	对于调用活动： BPC.BFM.ACTIVITY.MESSAGE。对于所有其他活动类型： BPC.BFM.ACTIVITY.STATUS	启动	ENTRY	活动已启动。对于调用活动，有可用的业务对象有效内容。
21011	对于调用活动、人员任务活动、接收活动和应答活动： BPC.BFM.ACTIVITY.MESSAGE。对于检取活动：BPC.BFM.ACTIVITY.EVENT。对于所有其他活动类型： BPC.BFM.ACTIVITY.STATUS	停止	EXIT	活动已完成。对于调用活动、人员任务活动、接收活动和应答活动，有可用的业务对象有效内容。
21021	BPC.BFM.ACTIVITY.STATUS	报告	DEASSIGNED	声明已取消。当人员任务活动的声明被取消时，将生成此事件。

代码	扩展名	情况	事件性质	描述
21022	BPC.BFM.ACTIVITY.CLAIM	报告	ASSIGNED	活动已声明。当声明人员任务活动时，将生成此事件。
21027	BPC.BFM.ACTIVITY.STATUS	停止	TERMINATED	活动已终止。长时间运行活动可能由于该活动被指定到的作用域或流程中进行的故障处理而终止。
21080	BPC.BFM.ACTIVITY.FAILURE	已失败	FAILED	活动已失败
21081	BPC.BFM.ACTIVITY.STATUS	报告	EXPIRED	活动已到期。可以对调用和直接插入人员任务活动定义到期时间。
42005	BPC.BFM.ACTIVITY.STATUS	报告	SKIPPED	活动已被跳过。此事件仅适用于定义了连接行为的活动。如果连接行为求值为 false，那么将跳过该活动并发出跳过事件。
42012	BPC.BFM.ACTIVITY.MESSAGE	报告	OUTPUTSET	活动输出消息已设置。有可用的业务对象有效内容。
42013	BPC.BFM.ACTIVITY.MESSAGE	报告	FAULTSET	活动故障消息已设置。有可用的业务对象有效内容。
42015	BPC.BFM.ACTIVITY.STATUS	停止	STOPPED	活动已停止
42031	BPC.BFM.ACTIVITY.STATUS	报告	FRETRIED	活动已被强制重试。要强制重试活动，请使用业务流程编排器资源管理器。
42032	BPC.BFM.ACTIVITY.STATUS	停止	FCOMPLETED	活动已被强制完成。要强制完成活动，请使用业务流程编排器资源管理器。
42036	BPC.BFM.ACTIVITY.MESSAGE	报告	EXIT	活动接收到消息
42037	BPC.BFM.ACTIVITY.STATUS	报告	CONDTRUE	循环条件为 true
42038	BPC.BFM.ACTIVITY.STATUS	报告	CONDFALSE	循环条件为 false
42039	BPC.BFM.ACTIVITY.WISTATUS	报告	WI_DELETED	工作项已删除。此事件仅适用于检索、直接插入人员任务以及接收事件。

代码	扩展名	情况	事件性质	描述
42040	BPC.BFM.ACTIVITY.WISTATUS	报告	WI_CREATED	工作项已创建。此事件仅适用于检取、直接插入人员任务以及接收事件。
42050	BPC.BFM.ACTIVITY.ESCALATED	报告	ESCALATED	活动已升级。此事件仅适用于检取、直接插入人员任务以及接收事件。
42054	BPC.BFM.ACTIVITY.WISTATUS	报告	WI_REFRESHED	活动工作项已刷新。此事件仅适用于检取、直接插入人员任务以及接收事件。
42055	BPC.BFM.ACTIVITY.WITRANSFER	报告	WI_TRANSFERRED	工作项已移交。此事件仅适用于检取、直接插入人员任务以及接收事件。
42057	BPC.BFM.ACTIVITY.FOREACH	报告	BRANCHES_STARTED	对于每个已启动的活动分支

对于活动事件而言，下列事件相关范围标识包含下列内容：

- *ECSCurrentID* 提供活动的标识。
- *ECSParentID* 提供外层流程的标识。

活动作用域事件

下表对所有活动作用域事件进行描述。

代码	扩展名	情况	事件性质	描述
42020	BPC.BFM.ACTIVITY.STATUS	启动	ENTRY	作用域已启动
42021	BPC.BFM.ACTIVITY.STATUS	报告	SKIPPED	作用域已被跳过
42022	BPC.BFM.ACTIVITY.FAILURE	失败	FAILED	作用域已失败
42023	BPC.BFM.ACTIVITY.STATUS	报告	FAILING	作用域正在终止
42024	BPC.BFM.ACTIVITY.STATUS	停止	TERMINATED	作用域已终止
42026	BPC.BFM.ACTIVITY.STATUS	停止	EXIT	作用域已完成
42043	BPC.BFM.ACTIVITY.STATUS	报告	COMPENSATING	作用域正在进行补偿
42044	BPC.BFM.ACTIVITY.STATUS	停止	COMPENSATED	作用域已进行补偿
42045	BPC.BFM.ACTIVITY.STATUS	失败	COMPFAILED	作用域补偿失败
42048	BPC.BFM.ACTIVITY.EVENT	报告	EV_RECEIVED	活动事件已接收
42051	BPC.BFM.ACTIVITY.ESCALATED	报告	EV_ESCALATED	作用域事件已升级

活动作用域事件是一种活动事件，上面有关 BPC.BFM.ACTIVITY.STATUS 的内容对此类事件的语法作了描述。

对于活动作用域事件而言，下列事件相关范围标识包含下列内容：

- ECSCurrentID 提供作用域的标识。
- ECSParentID 提供外层流程的标识。

链接事件

下表对所有链接事件进行描述。

代码	扩展名	情况	事件性质	描述
21034	BPC.BFM.LINK.STATUS	报告	CONDTRUE	链接求值为 true
42000	BPC.BFM.LINK.STATUS	报告	CONDFALSE	链接求值为 false

对于链接事件而言，下列事件相关范围标识包含下列内容：

- ECSCurrentID 提供链接的源活动标识。
- ECSParentID 提供外层流程的标识。

变量事件

下表对变量事件进行描述。

代码	扩展名	情况	事件性质	描述
21090	BPC.BFM.VARIABLE.STATUS	报告	CHANGED	变量更新。有可用的业务对象有效内容。

对于变量事件而言，下列事件相关范围标识包含下列内容：

- ECSCurrentID 提供外层流程的标识。
- ECSParentID 是当前流程的流程实例启动事件发生前的 ECSCurrentID 值。

业务流程事件中的情况

可以在不同的情况下发出业务流程事件。情况元素对这些情况的数据进行描述。

业务流程事件可以包含下列其中一个情况元素。

情况名	公共基本事件的内容	
启动	categoryName 设置为 StartSituation。	
	situationType	
	Type	StartSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	START_COMPLETED

情况名	公共基本事件的内容	
停止	categoryName 设置为 StopSituation。	
	situationType	
	Type	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	STOP_COMPLETED
破坏	categoryName 设置为 DestroySituation。	
	situationType	
	Type	DestroySituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
失败	categoryName 设置为 StopSituation。	
	situationType	
	Type	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	UNSUCCESSFUL
	situationQualifier	STOP_COMPLETED
报告	categoryName 设置为 ReportSituation。	
	situationType	
	Type	ReportSituation
	reasoningScope	EXTERNAL
	reportCategory	STATUS

第 20 章 监视人员任务事件

为人员任务发出的事件包含与情况无关的数据以及特定于人员任务事件的数据。下面对特定于人员任务事件的属性和元素进行描述。

人员任务事件的事件内容分为下列各类。

人员任务所特有的事件数据

事件是为任务和升级创建的。下面对每一类事件中特定于对象的内容进行描述。

对于业务流程编排器 V6.1 而言，有两种事件格式：

WebSphere Business Monitor 6.0.2 格式

如果存在 WebSphere Integration Developer 6.0.2 中设计的任务，或者在 WebSphere Integration Developer 6.1 中启用了 WebSphere Business Monitor 6.0.2 格式方式，那么将发生 WebSphere Business Monitor 6.0.2 格式的事件。除非另外指定，否则这些事件中特定于对象的内容将作为字符串类型的 XML 元素 *extendedDataElement* 而被写入。

WebSphere Business Monitor 6.1 格式

如果存在 WebSphere Integration Developer 6.1 中设计的任务，并且已启用 WebSphere Business Monitor 6.1 格式方式，那么将发生 WebSphere Business Monitor 6.1 格式的事件。这些事件中特定于对象的内容将作为 XML 元素写入公共基本事件中 eventPointData 文件夹中的 *xs:any* 槽。XML 的结构在 XML 模式定义 (XSD) 文件 *HTMEvents.xsd* 中定义。此文件在 *install_root\ProcessChoreographer\client* 目录中。

任务

任务事件包含下列特定于对象的事件内容。

属性	描述
taskTemplateName	派生实例的任务模板的名称。
taskTemplateValidFrom	模板的生效日期。
taskTemplateId	派生实例的任务模板的标识。
taskInstanceDescription	任务实例在缺省语言环境中的描述。
PayloadType	有效内容类型。此字符串值可以是 none、digest 或 full。此值取决于 WebSphere Integration Developer 中的设置。如果没有可用的显式监视定义，那么缺省有效内容类型为 full。

升级

升级包含下列特定于对象的事件内容：

属性	描述
taskTemplateName	派生实例的任务模板的名称。

属性	描述
taskTemplateValidFrom	模板的生效日期。
taskTemplateId	派生实例的任务模板的标识。
escalationName	升级的名称。
escalationInstanceDescription	可选：升级实例的描述。
PayloadType	有效内容类型。此字符串值可以是 none、digest 或 full。此值取决于 WebSphere Integration Developer 中的设置。如果没有可用的显式监视定义，那么缺省有效内容类型为 full。

人员任务事件的扩展名

扩展名指示人员任务事件的有效内容。本主题列示人员任务事件的所有扩展名及其相应有效内容。

扩展名包含一个字符串值，此字符串值用作公共基本事件的 *extensionName* 属性值。此字符串值也是用于提供其他事件数据的 XML 元素的名称。事件元素的名称是大写的，例如 BPC.HTM.BASE。XML 元素的名称是混合大小写的，例如 *HTMEventCode*。除非另有指示，否则所有数据元素的类型都是 string。

人员任务事件可以使用下列扩展名：

- 『 BPC.HTM.BASE 』
- 第 535 页的 『 BPC.HTM.TASK.BASE 』
- 第 535 页的 『 BPC.HTM.TASK.STATUS 』
- 第 535 页的 『 BPC.HTM.TASK.FOLLOW 』
- 第 535 页的 『 BPC.HTM.TASK.MESSAGE 』
- 第 536 页的 『 BPC.HTM.TASK.INTERACT 』
- 第 536 页的 『 BPC.HTM.TASK.FAILURE 』
- 第 536 页的 『 BPC.HTM.TASK.WISTATUS 』
- 第 536 页的 『 BPC.HTM.TASK.WITRANSFER 』
- 第 536 页的 『 BPC.HTM.ESCALATION.STATUS 』
- 第 537 页的 『 BPC.HTM.ESCALATION.WISTATUS 』
- 第 537 页的 『 BPC.HTM.ESCALATION.WITRANSFER 』

BPC.HTM.BASE

BPC.HTM.BASE 继承 WBIMonitoringEvent 的 XML 元素。

表 79. BPC.HTM.BASE 的 XML 元素

XML 元素	描述
<i>HTMEventCode</i>	业务流程编排器事件码，用于标识事件类型的编号。下列各表列示了可能的事件码。

BPC.HTM.TASK.BASE

BPC.HTM.TASK.BASE 继承第 534 页的『BPC.HTM.BASE』的 XML 元素。

表 80. BPC.HTM.TASK.BASE 的 XML 元素

XML 元素	描述
<i>taskTemplateId</i>	模板的标识。
<i>taskTemplateName</i>	任务模板的名称。此名称可以与显示名不同。
<i>taskTemplateValidFrom</i>	任务模板的生效日期和时间。

BPC.HTM.TASK.STATUS

BPC.HTM.TASK.STATUS 继承『BPC.HTM.TASK.BASE』的 XML 元素。

表 81. BPC.HTM.TASK.STATUS 的 XML 元素

XML 元素	描述
<i>taskInstanceDescription</i>	任务的描述。

BPC.HTM.TASK.FOLLOW

BPC.HTM.TASK.FOLLOW 继承『BPC.HTM.TASK.BASE』的 XML 元素。

表 82. BPC.HTM.TASK.FOLLOW 的 XML 元素

XML 元素	描述
<i>followTaskId</i>	作为后续任务启动的任务的标识。

BPC.HTM.TASK.MESSAGE

BPC.HTM.TASK.MESSAGE 继承『BPC.HTM.TASK.STATUS』的 XML 元素。

表 83. BPC.HTM.TASK.FOLLOW 的 XML 元素

XML 元素	描述
<i>message</i> 或 <i>message_BO</i>	<p>包含输入消息或输出消息的字符串或业务对象表示。格式取决于是否在 WebSphere Integration Developer 的事件监视器选项卡上选中监视兼容的事件选项。</p> <p>此属性仅适用于 WebSphere Business Monitor 6.0.2 格式的事件。对于 WebSphere Business Monitor 6.1 格式的事件而言，消息内容将写入 <i>applicationData</i> 部分，此部分包含一个名称设置为消息名称的内容元素。</p>

BPC.HTM.TASK.INTERACT

BPC.HTM.TASK.INTERACT 继承第 535 页的『BPC.HTM.TASK.STATUS』的 XML 元素。

表 84. BPC.HTM.TASK.INTERACT 的 XML 元素

XML 元素	描述
<i>username</i>	对于 BPC.HTM.TASK.INTERACT 而言, 这是与任务相关联的用户的名称。

BPC.HTM.TASK.FAILURE

BPC.HTM.TASK.FAILURE 继承第 535 页的『BPC.HTM.TASK.STATUS』的 XML 元素。

表 85. BPC.HTM.TASK.FAILURE 的 XML 元素

XML 元素	描述
<i>taskFailedException</i>	包含由分号 (;) 分隔的 <i>faultNameSpace</i> 和 <i>faultName</i> 的字符串。

BPC.HTM.TASK.WISTATUS

BPC.HTM.TASK.WISTATUS 继承第 535 页的『BPC.HTM.TASK.STATUS』的 XML 元素。

表 86. BPC.HTM.TASK.WISTATUS 的 XML 元素

XML 元素	描述
<i>username</i>	对于 BPC.BPC.TASK.WISTATUS 而言, 这是已创建或删除工作项的用户的列表。

BPC.HTM.TASK.WITRANSFER

BPC.HTM.TASK.WITRANSFER 继承第 535 页的『BPC.HTM.TASK.STATUS』的 XML 元素。

表 87. BPC.HTM.TASK.WITRANSFER 的 XML 元素

XML 元素	描述
<i>current</i>	对于 BPC.HTM.TASK.WITRANSFER 而言, 这是当前用户的名称。这是已将工作项移交他人的用户。
<i>target</i>	对于 BPC.HTM.TASK.WITRANSFER 而言, 这是工作项接收者的用户名。

BPC.HTM.ESCALATION.STATUS

BPC.HTM.ESCALATION.STATUS 继承第 535 页的『BPC.HTM.TASK.BASE』的 XML 元素。

表 88. *BPC.HTM.ESCALATION.STATUS* 的 XML 元素

XML 元素	描述
<i>escalationName</i>	升级的名称。
<i>escalationInstanceDescription</i>	升级的描述。

BPC.HTM.ESCALATION.WISTATUS

BPC.HTM.ESCALATION.WISTATUS 继承第 536 页的『BPC.HTM.ESCALATION.STATUS』的 XML 元素。

表 89. *BPC.HTM.ESCALATION.WISTATUS* 的 XML 元素

XML 元素	描述
<i>username</i>	对于 BPC.HTM.ESCALATION.WISTATUS 而言，这是已将工作项升级的用户的列表。

BPC.HTM.ESCALATION.WITRANSFER

BPC.HTM.ESCALATION.WITRANSFER 继承第 536 页的『BPC.HTM.ESCALATION.STATUS』的 XML 元素。

表 90. *BPC.HTM.ESCALATION.WITRANSFER* 的 XML 元素

XML 元素	描述
<i>current</i>	对于 BPC.HTM.ESCALATION.WITRANSFER 而言，这是当前用户的名称。这是已将工作项移交他人的用户。
<i>target</i>	对于 BPC.HTM.ESCALATION.WITRANSFER 而言，这是工作项接收者的用户名。

人员任务事件

如果您要求对 WebSphere Integration Developer 中的任务元素进行监视，那么将发送人员任务事件。下面列示了人员任务可以发出的所有事件。

当任务状态更改时，将发送事件。人员任务可能会引起下列类型的事件：

- 第 538 页的『任务事件』
- 第 539 页的『升级事件』

注： 仅当在任务模型中将业务相关性标志设置为 `true` 时，才会对临时任务发出事件。

直接插入任务的事件作为活动事件发出。要获取这些事件的列表，请参阅第 525 页的『业务流程事件』。

XML 模式定义 (XSD) 文件

XML 模式定义 (XSD) 文件 `HTMEvents.xsd` 描述了事件结构。此文件在 `install_root\ProcessChoreographer\client` 目录中。

各个表列的关键点

下列各表中的列包含:

代码 包含事件的编号。对于 WebSphere Business Monitor 6.0.2 格式的事件而言, 此值将作为名为 *HTMEventCode* 的扩展数据元素写入公共基本事件。对于 WebSphere Business Monitor 6.1 格式的事件而言, 此值将写入公共基本事件的 *xs:any* 槽。

扩展名 包含一个字符串值, 此字符串值用作公共基本事件的 *extensionName* 属性值。

如果使用了 WebSphere Business Integration Modeler 来创建底层任务模型, 那么有效内容中包含消息数据的事件的扩展名可以由 # 字符以及后跟的附加字符进行扩展。这些附加字符用于对包含不同消息对象的公共基本事件进行区分。发出消息数据的事件还包含附加的嵌套 *extendedDataElement*, 以报告该数据对象的内容。有关更多信息, 请参阅 WebSphere Business Integration Modeler 的文档。

情况 引用人员任务事件的情况名。有关情况的详细信息, 请参阅第 539 页的『人员任务事件中的情况』。

事件性质

指向 *EventNature* 参数中业务流程元素事件情况的指针, 这些事件情况将显示在 WebSphere Integration Developer 中。

任务事件

下表对所有任务事件进行描述。

代码	扩展名	情况	事件性质	描述
51001	BPC.HTM.TASK.INTERACT	报告	CREATED	任务已创建
51002	BPC.HTM.TASK.STATUS	破坏	DELETED	任务已删除
51003	BPC.HTM.TASK.STATUS	启动	ENTRY	任务已启动
51004	BPC.HTM.TASK.STATUS	停止	EXIT	任务已完成
51005	BPC.HTM.TASK.STATUS	报告	DEASSIGNED	声明已取消
51006	BPC.HTM.TASK.INTERACT	报告	ASSIGNED	任务已声明
51007	BPC.HTM.TASK.STATUS	停止	TERMINATED	任务已终止
51008	BPC.HTM.TASK.FAILURE	失败	FAILED	任务已失败
51009	BPC.HTM.TASK.STATUS	报告	EXPIRED	任务已到期
51010	BPC.HTM.TASK.STATUS	报告	WAITFORSUBTASK	正在等待子任务
51011	BPC.HTM.TASK.STATUS	停止	SUBTASKCOMPLETED	子任务已完成
51012	BPC.HTM.TASK.STATUS	报告	RESTARTED	任务已重新启动
51013	BPC.HTM.TASK.STATUS	报告	SUSPENDED	任务已暂挂
51014	BPC.HTM.TASK.STATUS	报告	RESUMED	任务已恢复
51015	BPC.HTM.TASK.FOLLOW	报告	COMPLETEDFOLLOW	任务已完成并且后续任务已启动
51101	BPC.HTM.TASK.STATUS	报告	UPDATED	任务属性已更新
51103	BPC.HTM.TASK.MESSAGE	报告	OUTPUTSET	输出消息已更新。有可用的业务对象有效内容。

代码	扩展名	情况	事件性质	描述
51104	BPC.HTM.TASK.MESSAGE	报告	FAULTSET	故障消息已更新。有可用的业务对象有效内容。
51201	BPC.HTM.TASK.WISTATUS	破坏	WI_DELETED	工作项已删除
51202	BPC.HTM.TASK.WISTATUS	报告	WI_CREATED	工作项已创建
51204	BPC.HTM.TASK.WITRANSFER	报告	WI_TRANSFERRED	工作项已移交
51205	BPC.HTM.TASK.WISTATUS	报告	WI_REFRESHED	工作项已刷新

对于任务事件而言，事件相关范围的下列标识包含下列内容：

- ESCcurrentID 提供任务实例的标识。
- ECSParentID 是任务实例事件发生前的 ECSCurrentID。

升级事件

下表对所有任务升级事件进行描述。

代码	扩展名	情况	事件性质	描述
53001	BPC.HTM.ESCALATION.STATUS	报告	ENTRY	升级已触发
53201	BPC.HTM.ESCALATION.WISTATUS	破坏	WI_DELETED	工作项已删除
53202	BPC.HTM.ESCALATION.WISTATUS	报告	WI_CREATED	工作项已创建
53204	BPC.HTM.ESCALATION.WITRANSFER	报告	WI_TRANSFERRED	升级已移交
53205	BPC.HTM.ESCALATION.WISTATUS	报告	WI_REFRESHED	工作项已刷新

对于任务事件而言，事件相关范围的下列标识包含下列内容：

- ESCcurrentID 提供升级的标识。
- ECSParentID 提供相关任务实例的标识。

人员任务事件中的情况

可以在不同的情况下发出人员任务事件。情况元素对这些情况的数据进行描述。

人员任务事件可以包含下列其中一个情况元素。

情况名	公共基本事件的内容	
启动	categoryName 设置为 StartSituation。	
	situationType	
	Type	StartSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	START_COMPLETED

情况名	公共基本事件的内容	
停止	categoryName 设置为 StopSituation。	
	situationType	
	Type	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	STOP_COMPLETED
破坏	categoryName 设置为 DestroySituation。	
	situationType	
	Type	DestroySituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
失败	categoryName 设置为 StopSituation。	
	situationType	
	Type	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	UNSUCCESSFUL
	situationQualifier	STOP_COMPLETED
报告	categoryName 设置为 ReportSituation。	
	situationType	
	Type	ReportSituation
	reasoningScope	EXTERNAL
	reportCategory	STATUS

第 6 部分 调整

第 21 章 调整业务流程

通过本任务提高业务流程的性能。

开始之前

成功运行业务流程后，可以执行本任务以提高性能。

过程

1. 定义如何度量基准性能以及需要优化哪些度量值。

例如，对于某些业务应用程序而言，最好在最大负载条件下缩短最终用户的响应时间。对于其他应用程序而言，系统处理事务的速度可能比每个事务的实际持续时间更为重要。

2. 进行基准度量。

在负载、时间和工作日等条件适于调整应用程序的情况下进行基准度量。通常，最重要的基准度量值是吞吐量和响应时间。吞吐量值是在达到 100% CPU 负载、最大磁盘 I/O 吞吐量或 100% 网络 I/O 吞吐量等特定瓶颈容量时测量的。最好在服务器使用率较低时针对单个流程实例进行度量，从而获得可靠的响应时间值。

3. 调整应用程序。

应用程序可以包含多个流程。由于微流程的性能比长时间运行流程好，因此，如果不必考虑持久性，并且可以在一个事务中以单线程方式处理功能，那么最好设计微流程。并且，您应该考虑将长时间运行流程的分支划分为微流程。此外，同步服务调用的执行速度通常比异步服务调用快。所以，为了提高性能，最好使用同步服务调用，尽管这不是长时间运行流程中的缺省行为。

在长时间运行流程中，可以更改事务边界。在大多数情况下，减少事务边界数有助于提高性能。但是，只能通过进行性能测试来确定最佳的事务边界数。此外，您应该考虑在流程中使用并行执行路径来代替将活动序列化，并应该考虑最大程度地降低流程中流动的数据的大小和复杂性，这是因为，对数据进行序列化和反序列化处理的成本很高。并且，应该最大程度地减少为了进行日志记录而发出的事件数。

4. 调整流程。

根据应用程序使用了长时间运行流程还是微流程，执行下列其中一个步骤：

- 要调整长时间运行流程，请执行第 544 页的『调整长时间运行流程』中描述的步骤。这些流程通常运行时间较长，但会被事件或人员交互中断。因此，它们的性能取决于业务流程编排器数据库和消息传递服务的性能。
 - 要调整微流程，请执行第 555 页的『调整微流程』中描述的步骤。这些流程的运行时间通常较短。它们使用数据库的目的仅仅是，在审计日志记录功能处于启用状态时执行此功能以及检索模板信息。它们不使用消息传递支持来存储持久数据。这些流程不涉及人员交互。
5. 审查可以消除的性能瓶颈的当前配置。

要考虑的可能性包括：

- 安装更多处理器、更多内存以及速度更快的磁盘。
 - 将数据库日志与数据存储在不同的物理磁盘上，并且将数据分布在多个磁盘上。
 - 使用 DB2 来代替 Cloudscape，以获得最佳性能。
6. 在类似于基准度量负载条件的负载条件下重复进行基准度量。

保留应用程序性能度量记录，以便有目的地度量将来的性能更改情况。

结果

已对业务流程进行配置，其运行速度明显加快。

调整长时间运行流程

通过本任务提高长时间运行业务流程的性能。

关于此任务

长时间运行流程可以包含用户交互、异步调用、多个接收活动、多个检取活动以及事件处理程序；它们使用数据库和消息传递子系统存储持久状态。下列主题描述如何提高长时间运行流程的性能。

平衡硬件资源

通过平衡硬件资源，可以提高长时间运行业务流程的性能。

关于此任务

开始对系统进行调整前，请验证所用计算机硬件的平衡状态是否良好，即 CPU、内存和 I/O 等可用资源之间是否具有恰当的比例关系。带有一个或多个速度非常快的 CPU 但内存很小或者 I/O 性能很差的计算机将难以进行调整。对于可中断的流程而言，多个高速磁盘驱动器所提供的良好 I/O 性能与足够的处理能力和充足的内存一样重要。

对于生产系统，建议您使用不同的机器来运行 DBMS 和应用程序服务器，从而将数据库与应用程序服务器分开。对于高负载或高可用性配置，请考虑使用基于多台机器的 WebSphere 集群来运行业务流程，并使用独立的机器来存储数据库。

过程

1. 在数据库机器上，请确保分配足够的磁盘。
2. 分配足够的内存。

要分配的内存量视平台而定：

- 对于带有 4 GB 物理内存和本地数据库管理系统的 32 位 Windows 系统，请按如下方式分配内存：
 - 为 Windows 系统分配 512 MB 内存
 - 为 WebSphere Application Server 分配 768 MB 内存
 - 如果正在使用 DB2，那么为数据库分配 1.5 GB 内存。如果正在使用 Oracle，那么为系统全局区域（SGA）分配不超过 1 GB 内存，为程序全局区域（PGA）分配 500 MB 内存。

- 对于带有 8 GB 物理内存和本地数据库管理系统的 64 位 AIX 系统，请按如下方式分配内存：
 - 为 AIX 系统分配 512 MB 内存
 - 为 WebSphere Application Server 分配 1024 MB 内存
 - 为数据库分配 5 GB 内存。为流程数据库分配 4 GB 内存，为消息传递数据库分配 1 GB 内存。

提示： 由于文件高速缓存等功能也要使用内存，因此，为了确保性能最佳，请不要将所有内存都分配给数据库。请避免必须将数据交换到磁盘的情况，在这种情况下，将没有足够的内存可用。

- 对于 i5/OS 系统，请使用使用系统状态（WRKSYSSTS）命令帮助防止系统进行内存页面调度。如果发生大量页面故障，那么请执行下列一项或多项操作：
 - a. 增加可供 WebSphere Process Server 或企业服务总线服务器子系统内存池使用的内存量
 - b. 将 WebSphere Process Server 或企业服务总线服务器移至另一个内存池
 - c. 从 WebSphere Process Server 或企业服务总线服务器子系统内存池中除去作业
- 对应用程序服务器的堆大小进行微调。

注： 如果应用程序服务器运行 i5/OS 系统，那么无法调整堆大小。

3. 观察网络利用率。应用程序的性能还取决于服务器与数据库服务器之间的消息传递速度。因此，应该尽可能缩短网络延迟。
4. 将工作负载移至其他服务器。

请考虑可以将哪些应用程序或子系统移至其他服务器。

结果

计算机硬件的平衡状态现已良好。

相关任务

第 95 页的『规划 BPEDB 数据库』
规划业务流程编排器的数据库。

第 549 页的『调整应用程序服务器』
通过本任务调整应用程序服务器。

指定 DB2 数据库初始设置

通过本任务指定 DB2 数据库的初始设置。请注意，本信息仅仅作作为示例提供。

关于此任务

警告： 以下信息与业务流程编排器数据库相关。有关调整 WebSphere 缺省消息传递数据库的信息，请参阅 WebSphere Application Server Network Deployment 信息中心中的调整消息传递引擎数据存储以及解决问题。

为了实现良好的数据库操作性能，请指定数据库的初始设置。此外，请使用两个具有不同条带大小的独立逻辑磁盘。并且，可以让每个实例使用一个数据库，以便消息传递引擎使用数据库作为数据存储时，它们可以使用业务流程编排器数据库进行消息传递，也可以使用独立的数据库机器来存放消息传递数据库。

注：此信息不适用于 DB2 UDB iSeries 版数据库，这是因为，这些设置是此数据库类型的内置设置。

过程

1. 将日志文件与数据文件分开。

如果有足够的磁盘驱动器可用，那么将数据库日志文件与数据存储在不同的磁盘驱动器上有助于提高性能。

例如，如果在 Windows 系统上使用 DB2，那么可以输入以下命令将数据库 BPEDB 的日志文件位置更改为 F:\db2logs 目录：

```
db2 UPDATE DB CFG FOR BPEDB USING NEWLOGPATH F:\db2logs
```

2. 创建表空间。

创建数据库后，必须明确地创建表空间。在 WebSphere Application Server 安装版本的 ProcessChoreographer 子目录中，业务流程编排器提供了用于创建表空间的示例脚本。定制这些脚本以满足特定情况的需求。创建表空间时，您的目标是将输入和输出操作分布到尽可能多可供 DB2 使用的磁盘驱动器上。缺省情况下，这些脚本将创建下列表空间：

AUDITLOG

包含流程和任务的审计跟踪表。根据所使用审计等级的不同，对此表空间中的表进行的访问量可能非常大。如果审计功能处于关闭状态，那么不会访问此表空间中的表。

COMP 包含业务流程编排器 V5 中业务流程的补偿表。根据可补偿流程和活动的百分比，此表空间中的表可能需要较高的磁盘带宽。如果在业务流程中未使用补偿功能，那么不会使用此表空间中的表。

INSTANCE

包含流程实例和任务的表。此表空间的使用率始终较高，而与所运行的长时间运行流程类型无关。有可能的话，请将此表空间分布到多个磁盘驱动器上。

SCHEDTS

包含 WebSphere 调度组件使用的表。由于在调度程序中使用了高速缓存机制，因此，对调度程序表空间中的表的访问量通常较低。

STAFFQRY

包含用于临时存储从诸如轻量级目录访问协议 (LDAP) 等人员注册表获取的人员查询结果的表。当业务流程包含许多人员活动时，将频繁地访问此表空间中的表。

TEMPLATE

包含用于存储流程模板信息和任务模板信息的表。将在应用程序部署期间填充这些表。在运行时，对这些表的访问频率并不高。除了在部署期间插入新数据以外，不会更新任何数据。

WORKITEM

包含工作项处理所需的表。工作项用于进行人员任务交互。根据业务流程中人员任务数的不同，对此表空间中的表的访问频率既可能很低，也可能非常高。由于还将生成工作项以支持长时间运行流程的管理工作，因此，即使未使用明确的人员任务，访问频率也不会为零。

要创建高性能的数据库，请执行下列操作：

a. 创建数据库。

在 Windows 上，可以指定目标驱动器。以下命令在目标驱动器上与服务器上缺省 DB2 实例同名的目录中创建一个数据库。例如，如果要在驱动器 D：上创建数据库，并且本地缺省实例是 DB2，那么数据库数据将存储在 D:\DB2 中。因此，要在 D：目录中为业务流程编排器创建 DB2 数据库，请输入以下命令：

```
CREATE DATABASE BPEDB ON D: USING CODESET UTF-8 TERRITORY en-us;
```

在 UNIX 和 Linux 上，输入以下命令：

```
CREATE DATABASE BPEDB ON /wasdbfs USING CODESET UTF-8 TERRITORY en-us;
```

其中 /wasdbfs 指定目录。

b. 在期望的磁盘上创建表空间。

例如，以下脚本使用 WebSphere Application Server 安装版本的 ProcessChoreographer 子目录中的 createTablespaceDb2.ddl 文件。此脚本将使用 Windows 系统上的单一高性能磁盘驱动器来创建表空间。

```
-- Scriptfile to create tablespaces for DB2 UDB
-- Replace occurrence of @location@ in this file with the location
-- where you want the tablespace containers to be stored, then run:
-- db2 connect to BPEDB
-- db2 -tf createTablespaceDb2.ddl
```

```
CREATE TABLESPACE TEMPLATE MANAGED BY SYSTEM USING( 'D:/BPE/TEMPLATE' );
CREATE TABLESPACE STAFFQRY MANAGED BY SYSTEM USING( 'D:/BPE/STAFFQRY' );
CREATE TABLESPACE AUDITLOG MANAGED BY SYSTEM USING( 'D:/BPE/AUDITLOG' );
CREATE TABLESPACE COMP MANAGED BY SYSTEM USING( 'D:/BPE/COMP' );
CREATE TABLESPACE INSTANCE MANAGED BY SYSTEM USING( 'D:/BPE/INSTANCE', 'D:/BPE/INSTANCE' );
CREATE TABLESPACE WORKITEM MANAGED BY SYSTEM USING( 'D:/BPE/WORKITEM' );
CREATE TABLESPACE SCHEDTS MANAGED BY SYSTEM USING( 'D:/BPE/SCHEDTS' );
```

c. 创建表。

通过运行产品提供的用于相关数据库的脚本，为业务流程编排器创建表。例如，对于 DB2，请使用 ProcessChoreographer 目录中的 createSchemaDb2.ddl 文件。

3. 调整数据库。

使用容量规划工具调整数据库的初始设置。

如果正在使用 DB2，那么通过从业务流程编排器数据库的弹出菜单中选择 **DB2 配置顾问程序**，从 DB2 控制中心启动 DB2 配置顾问程序。执行下列操作：

a. 为 DB2 分配内存。

对于服务器，仅向 DB2 分配不需要进行磁盘交换情况下的最大可用物理内存量。

b. 指定工作负载类型。

对于**工作负载**，选择**混合**，即包括查询和事务。

- c. 对于**事务数**，指定事务长度以及估计每分钟处理的事务数。

选择**多于 10** 个以指示使用长事务。

然后，在**每分钟事务数**字段中，选择估计每分钟处理的事务数。要确定此数目，请假定流程中每个活动各有一个事务。那么，每分钟执行的事务数如下：

$$\text{每分钟执行的事务数} = \text{每分钟完成的流程数} * \text{每个流程中的活动数}$$

- d. 调整数据库，以提高事务性能并降低恢复速度。

对于**优先级**，选择**事务性能较高**。

- e. 如果可能的话，请调整用产品中一般数据量填充的数据库。对于**已填充**，选择**是**。如果数据库中没有数据，那么选择**否**。
- f. 调整并行连接数设置。

对于**连接数**，指定可以对应用程序服务器建立的最大并行连接数。用于确定此值的准则如下所示：

- 所需的数据库连接数由连接到 WebSphere Application Server 的 Java 数据库连接 (JDBC) 数确定。JDBC 连接由 WebSphere Application Server 中的 JDBC 连接池提供。对于 p 个 JDBC 连接，需要 $p * 1.1$ 个数据库连接。第 549 页的『调整应用程序服务器』描述了如何估算 p 的实际值。
- 如果业务流程编排器与数据库安装在同一台物理服务器上，那么业务流程编排器不需要远程数据库连接。但是，由于可能需要使用远程连接来管理远程数据库，所以，请指定较小的非零值。
- 如果业务流程编排器与 DB2 安装在不同的服务器上，那么根据先前对本地连接描述的规则来设置远程应用程序数。

- g. 对于**隔离**，选择**读稳定性**。此隔离级别是业务流程编排器所必需的。

配置顾问程序将显示建议的更改。您可以立即应用更改，也可以将更改保存在文件中以便稍后应用。

结果

在当前环境和负载条件下，长时间运行流程已按最高速度运行。

指定 Oracle 数据库初始设置

要提高 Oracle 数据库的性能和可伸缩性，主要措施是优化数据库文件布局、为缓冲区高速缓存分配足够的内存以便高效地进行高速缓存以及调整数据库参数。

过程

1. 为缓冲区高速缓存分配足够的内存。

通过使用内存高速缓存，可以缩短数据库访问响应时间。这意味着，缓冲区高速缓存必须足够大。请将缓冲区高速缓存大小设置为至少 700 MB，然后监视高速缓存使用情况并在必要时增大高速缓存大小。

2. 调整日志文件大小以减少日志文件切换

Oracle 实例的事务日志保存在多个以循环方式使用的文件中。活动日志文件将在一个日志文件变满时进行切换，从而对最后一个活动日志进行存档。由于日志文件切换操作的成本高昂，因此，请调整日志文件大小以降低这些切换操作的发生频率；最好先指定初始值 750 MB。然后，请监视事务速率和平均日志大小，并根据需要调整此值。

3. 调整下列数据库参数。

UNDO_TABLESPACE

确保所使用的撤销表空间未超过其大小限制的 70%。

OPEN_CURSORS

此参数的缺省值是 50。但是，此值通常不够大。此参数可以使用的最大值随操作系统的不同而有所变化。在大多数操作系统上，支持高达 1000 的值。

MAX_SHARED_SERVERS

指定可以同时运行的最大共享服务器进程数。使用此参数为其他进程（例如专用服务器）保留进程槽。如果已指定 MAX_SHARED_SERVERS 参数值，那么它应该大于或等于 SHARED_SERVERS 参数值并小于 PROCESSES 参数值。例如，如果有 150 个并发用户，那么此参数的初始值最好是 MAX_SHARED_SERVERS=70。

规划消息传递引擎设置

通过本任务规划消息传递引擎的初始设置。

关于此任务

为了最大程度地提高长时间运行流程的性能，请对消息传递系统进行调整，以便最大程度地提高持久消息的性能。对于数据存储器后端类型而言，首选性能优良的文件存储器。如果环境在集群中运行，并且无法使用文件存储器，那么请使用数据库数据存储器。

如果要使用 WebSphere Application Server 的服务集成功能，那么请按照 WebSphere Application Server Network Deployment 信息中心中为服务集成设置调整属性提供的指示信息执行操作，以设置和调整消息传递引擎的数据存储器。

结果

消息传递引擎的运行性能已最佳。

相关任务

第 104 页的『规划消息传递引擎数据库』

通过让业务流程编排器总线的消息传递引擎使用独立的数据库，可以提高性能。

调整应用程序服务器

通过本任务调整应用程序服务器。

开始之前

开始执行本任务前，必须指定数据库的初始设置，如第 545 页的『指定 DB2 数据库初始设置』所述。

关于此任务

要确保业务流程容器的执行性能最佳，需要调整服务器设置。

过程

1. 估算每个业务流程容器所需的应用程序服务器资源。
 - a. 一个数据源，用于对数据库 BPEDDataSourceDb2 读写业务流程状态信息，此数据库在服务器作用域 DB2 通用 JDBC 驱动程序提供程序 (XA) 中
 - b. 通过对下列数目进行累加，计算用于进行流程导航的最大并发事务数 t :
 - 通过业务流程编排器 API 并发连接的最大客户机数
 - JMS 激活规范 BPEInternalActivationSpec 中定义的并发端点数
 - JMS 激活规范 HTMInternalActivationSpec 中定义的并发端点数要查看流程服务器的激活规范，请在管理控制台中单击资源 → JMS 提供程序 → 缺省消息传递 → 激活规范。
 - c. 对于业务流程编排器数据库，请计算所需的并行 JDBC 连接数，即 $p = 1.1 * t$
 p 值不能大于数据库所允许的连接数。
 - d. 对于消息传递数据库，请计算所需的并行 JDBC 连接数，即 $m = t + x$ ，其中 x 是为了应对必须处理所生成的附加消息这种超负荷情况而需使用的附加 JMS 会话数
 - e. 将 SQL 语句高速缓存大小设置为 50。
2. 调整业务流程编排器数据库 BPEDB 的 JDBC 提供程序设置。
 - a. 将**最大连接数**设置为 p 值。 p 值不能大于数据库所允许的连接数。
 - b. 将 SQL 语句高速缓存大小设置为 300。
3. 重复执行步骤 2，以调整 SCA 消息传递引擎数据存储器的 JDBC 提供程序设置。
4. 调整消息传递数据库的 JDBC 提供程序设置。

将**最大连接数**设置为 m 值。

5. 调整堆大小。

下面是 32 位系统上服务器堆大小的一些准则。这些准则不适用于运行 i5/OS 系统的服务器。

- 256 MB 太小，此值将导致性能不佳。
- 对于许多系统而言，512 MB 作为初始堆大小已足够。
- 1024 MB 是合理的上限。

对于 64 位系统，堆的合理大小为 1 到 2 GB。

6. 调整业务流程使用的服务。确保调整支持服务，以与并发度及业务流程编排器对该服务的负载需求相匹配。

结果

为提高性能，已经对应用程序服务器进行了调整。

相关任务

第 544 页的『平衡硬件资源』

通过平衡硬件资源，可以提高长时间运行业务流程的性能。

对数据库进行微调

通过本任务对数据库进行微调。

开始之前

业务流程容器和业务流程必须处于运行状态。

关于此任务

注：如果您未使用 DB2，那么请参阅数据库管理系统文档，以了解有关监视数据库性能、确定和消除瓶颈以及对数据库性能进行微调的信息。本主题其余内容中提供的建议适用于 DB2 用户。但是，此信息不适用于 DB2 UDB iSeries 版。

过程

1. 根据缓冲池的用途和命中率指定其大小

缓冲池命中率是指可从池中已包含的数据获得应答的数据库请求所占的百分比。此值应该接近 100%，但任何高于 90% 的值都可接受。请增大缓冲池的 **SIZE** 参数值，直到命中率令人满意为止。您应该监视内存分配总量。如果缓冲池设置得过大，那么系统将开始进行页面交换。在这种情况下，请减小缓冲池大小或者增加可用内存。

如果正在使用 DB2 版本 8，那么可以计算缓冲池命中率。可以从缓冲池快照中获取进行计算所需的值。请使用以下命令来获取快照：

```
DB2 get snapshot for all bufferpools
```

有关计算命中率的更多信息，请参阅 DB2 V8 信息中心。

如果正在使用 DB2 版本 9，那么请使用 BP_HITRATIO 管理视图来检索命中率信息。有关此视图的更多信息，请参阅 DB2 V9 信息中心。

DB2 配置顾问程序能够提供建议的缓冲池大小值，在缺省情况下，业务流程编排器数据库仅使用 IBMDEFAULTBP 缺省缓冲池。您可以使用以下命令来设置此缓冲池的大小：

```
DB2 ALTER BUFFERPOOL IBMDEFAULTBP SIZE 120000
```

此命令将显示缓冲池及其大小（以页数计）以及每一页的大小：

```
DB2 select BPNAME, NPAGES, PAGESIZE from syscat.bufferpools
```

2. 调整锁定列表空间，以帮助确保性能最佳。

所有锁定都需要存储器，此存储器大小有限。如果事务请求使用的锁定超出此限制，那么这些事务将异常中止，从而导致性能下降。

a. 检查 DB2 实例的 db2diag.log 文件。

查找与以下示例类似的条目：

```
2005-07-24-15.53.42.078000 实例: DB2 节点: 000  
PID:2352(db2syscs.exe) TID:4360 Appid:*LOCAL.DB2.027785142343  
数据管理 sqlEscalateLocks 探针: 4 数据库: BPEDB
```

```
ADM5503E 将表"DB2ADMIN.ACTIVITY_INSTANCE_B_T"上的"10"个锁定升级为锁定意图"X"  
时失败。  
SQLCODE 为"-911"。
```

此类消息表示已超出锁定空间限制。

b. 增大 MAXLOCKS 和 LOCKLIST 参数值。

这些参数控制数据库的锁定升级行为。锁定升级将同一个表中多个独立的行级锁定转换为一个表级锁定。

如果某个事务使用的锁定数超出锁定列表的 MAXLOCKS 参数值，那么数据库管理器将把这些锁定转换为一个表锁定，以避免超出锁定空间限制。但是，锁定升级将大大增加死锁机会。因此，请将 MAXLOCKS 参数值增大到 60%。

请将 LOCKLIST 参数值增大到约为 $10 * p$ ，其中 p 是任意时刻所需的最大并行 JDBC 连接数估算值。例如，如果已对业务流程编排器数据库 BPEDB 的大小进行调整，并且 p 值为 50，那么请输入以下命令：

```
db2 UPDATE DB CFG FOR BPEDB USING LOCKLIST 500
```

3. 如果您曾使用 DB2 配置顾问程序，那么数据库吞吐量可能已令人非常满意。但是，可以通过下列方法进一步提高性能：

- 按照 DB2 联机文档、书籍和文章中描述的数据库调整最佳做法执行操作。
- 调整下列 DB2 参数：

AVG_APPLS

将此参数设置得太大比设置得太小好。例如，如果最多有 20 个已连接的应用程序，那么请将 AVG_APPLS 设置为 50。

DLCHKTIME

此参数指定死锁检测时间范围。缺省值为 10 秒。

LOCKTIMEOUT

此参数指定应用程序等待锁定的时间。缺省值为 -1，这表示应用程序将等待到获取锁定或者发生死锁为止。此参数的值应该始终大于 DLCHKTIME 参数值，以便将死锁报告为死锁而非锁定超时。最好将此参数的初始值设置为 30 秒。如果负载测试表明事务时间将超过 30 秒，那么应该设置更大的值。

LOGBUFSZ

如果增大 DB2 日志缓冲区大小，那么可以降低必须将已满的日志缓冲区写入磁盘的频率。

LOG_FILSZ

如果增大日志文件的大小，那么可以降低日志文件的切换频率。

4. 根据工作负载需求，调整数据库设置和数据库管理器设置。使用配置顾问程序配置数据库后，还可以调整下列设置：

MINCOMMIT

强烈建议您使用值 1。DB2 配置顾问程序可能会建议您使用其他值。

NUM_IOCLEANERS

对于只执行查询的应用程序，请将此值设置为 0。对于常规处理，请使用介于 1 与系统中磁盘驱动器数目的值（另请参阅 NUM_IOSERVERS 参数）。对于大型缓冲池而言，通常最好使用较大的数目。

NUM_IOSERVERS

必须与用于存储数据库的物理磁盘数匹配。IO 服务器数应该至少与磁盘数相同。IO 服务器不会使用很多系统资源，因此，设置太大的值比设置太小的值好。

5. 更新数据库的 DB2 统计信息。

在系统上有了初始负载之后，或者每当数据库中的数据量发生显著变化时，请考虑更新包含统计信息的 DB2 系统目录表。请使用 RUNSTATS 命令来更新统计信息。

最好使用脚本来运行 RUNSTATS 命令。以下示例说明这样的脚本。此示例假定您以 bpeuser 用户身份使用密码 password 登录，并且将连接到业务流程编排器数据库 BPEDB。DB2 命令将生成一个 Windows 命令文件，此文件将更新 BPEDB 数据库中相关表空间中所有表的统计信息。对于 TEMPLATE 表空间中的表而言，由于不会频繁地访问或更新这些表中的信息，因此省略了这些表。

```
db2 -x "select 'db2 runstats on table '
        concat rtrim(tabschema)
        concat '.'
        concat tablename
        concat ' with distribution and detailed indexes all '
from syscat.tables
where
        type='T' AND
        TBSPACEID IN (select TBSPACEID from sysibm.systablespace
        where TBSPACE IN ('INSTANCE', 'WORKITEM', 'STAFFQUERY',
        'AUDITLOG', 'SCHEDTS'))" > runStatsScript.cmd

echo db2 connect reset >> runStatsScript.cmd
```

生成的命令文件将更新所指定表的统计信息。它包含类似于以下的条目：

```
db2 runstats on table BPEUSER.ACTIVITY_INSTANCE_B_T with distribution and
detailed indexes all
db2 runstats on table BPEUSER.AUDIT_LOG_T with distribution and
detailed indexes all
...
db2 connect reset
```

您可能想对此命令文件进行扩展，以便在调用 RUNSTATS 命令前运行 REORG 命令。有关如何使用 REORG 命令来重组数据库表的信息，请参阅 DB2 文档。

6. 避免死锁。

当至少两个事务相互阻塞对方对资源的访问时，将发生死锁。数据库配置不佳会引起死锁。使用业务流程编排器 API 的方式也会导致死锁。为了避免死锁，每个 API 调用或者数据库对象查询操作都应该在自己的事务中运行。

业务流程管理器可以从数据库死锁中恢复。但是，因为从检测到死锁到重试已回滚事务之间需要经过一段时间，所以死锁会对性能产生重大影响。因此，为了提高性能，应该避免死锁。

要检查死锁，请检查 db2diag.log 文件并使用 DB2 监视器。

a. 提高 db2diag.log 文件的记录级别，以获取更多有关数据库瓶颈的信息。

将 DIAGLEVEL 参数值由 3（缺省值）增大到 4，以包括错误消息、警告消息和参考消息。可以使用以下命令来更改此值：

```
db2 update dbm cfg using DIAGLEVEL 4
```


b. 创建 DB2 事件监视器。

事件监视器能够提供有关某些事件（例如死锁）的更多信息。

1) 使用以下命令来创建事件监视器:

```
db2 create event monitor monitor_name for statements, connections,  
transactions, deadlocks with details write to file file_name
```

2) 使用以下命令来启动事件监视器:

```
db2 set event monitor monitor_name state=1
```

3) 使用以下命令来收集信息:

```
db2evmon -db database_name -evm monitor_name output_file_name
```

c. 使用数据库快照监视器来收集统计信息。

快照监视器使用数据库监视开关。对于数据库实例而言，监视开关的缺省设置如下所示:

缓冲池 (DFT_MON_BUFPOOL) = ON

锁定 (DFT_MON_LOCK) = ON

排序 (DFT_MON_SORT) = OFF

语句 (DFT_MON_STMT) = OFF

表 (DFT_MON_TABLE) = OFF

时间戳记 (DFT_MON_TIMESTAMP) = ON

工作单元 (DFT_MON_UOW) = OFF

要查看数据库实例的当前设置，请使用以下命令并搜索所有以 DFT_MON_ 开头的参数:

```
db2 get dbm cfg
```

这些设置与数据库的相应设置不同。对于数据库而言，监视记录开关的缺省设置如下所示:

数据库分区 0 的开关列表

缓冲池活动信息 (BUFFERPOOL) = ON

锁定信息 (LOCK) = ON

排序信息 (SORT) = OFF SQL

语句信息 (STATEMENT) = OFF

表活动信息 (TABLE) = OFF

获取时间戳记信息 (TIMESTAMP) = ON

工作单元信息 (UOW) = OFF

要查看数据库的当前设置，请使用以下命令:

```
db2 get monitor switches
```

- 要更新某个数据库监视器（例如锁定监视器）的设置，请使用以下命令:

```
db2 update monitor switches using lock on
```

此设置仅对当前数据库会话有效。

- 要更新数据库实例的某个监视开关（例如锁定监视器）的设置，请使用以下命令:


```
db2 update dbm cfg using DFT_MON_LOCK OFF
```

要激活此设置，请重新启动数据库实例。

- 在启用快照监视器前，请使用以下命令复位计数器：

```
db2 reset monitor all
```

- 重新启动数据库实例后，要获取当前快照，请使用以下命令：

```
db2 get snapshot for all on database_name output_file_name
```

结果

在当前环境和负载条件下，长时间运行流程已按最高速度运行。

对消息传递提供程序进行微调

通过本任务提高消息传递提供程序的性能。

过程

如果使用 WebSphere Application Server 的服务集成功能，那么请参阅 WebSphere Application Server 信息中心中的调整消息传递引擎数据存储库以及解决问题。

结果

消息传递提供程序的性能已提高。

调整微流程

通过本任务提高微流程的性能。

关于此任务

微流程在内存中运行，无需与用户进行交互，也不需要持久消息传递支持。仅当对微流程启用了审计日志记录或公共事件基础结构（CEI）时，才需要访问数据库。微流程的处理在单一线程中进行，并且通常在单个事务中完成。微流程的性能主要取决于所调用的服务。但是，如果服务器的可用内存量太少，微流程的性能将会下降。

过程

1. 调整 Java 虚拟机（JVM）堆大小。

由于较大的堆大小将减少所需的垃圾回收周期数，因此，通过增大 Java 堆大小，可以提高微流程的吞吐量。请保持此值足够小，以避免堆内容被交换到磁盘上。有关服务器堆大小的准则，请参阅第 549 页的『调整应用程序服务器』中的相关步骤。

2. 调整 JVM 垃圾回收。使用“吞吐量垃圾回收器”能够实现最高吞吐量，但根据堆大小的不同，垃圾回收暂停时间可能长达 100-1000 毫秒。如果响应时间比吞吐量更重要，请使用“低暂停时间垃圾回收器”。
3. 调整对象请求代理程序（ORB）线程池大小。如果远程客户机连接到服务器端 ORB，请确保 ORB 线程池中有足够的可用线程。
4. 调整缺省线程池大小。要增加可以并发运行的微流程数，必须增大缺省线程池大小。要更改此值，请在管理控制台中单击服务器 → 应用程序服务器 → **server** → 添加属性 → 线程池 → 缺省值。

结果

在当前环境和负载条件下，微流程已按最高速度运行。

对包含人员任务的业务流程进行调整

对于包含人员任务的业务流程，可以通过多种方法提高其性能。

下列主题描述如何对包含人员任务的业务流程进行调整。

减少对人员任务进行的并发访问

当两位以上人员尝试声明同一个人任务时，只有一位人员能够成功地完成任务。其他人员的访问请求将被拒绝。

一个人任务只能被一位人员声明。如果多位人员尝试同时处理同一个人任务，那么将增大发生冲突的可能性。发生冲突时，由于要等待锁定数据库或者执行回滚，因此会导致延迟，某些用于避免冲突或减少冲突机会的方法如下所示：

- 如果并发访问量较高，请对能够访问特定人员任务的用户数进行限制。
- 通过使用智能声明机制，避免客户机执行不必要的人员任务查询操作。例如，可以执行下列其中一个步骤：
 - 如果第一次声明不成功，那么尝试声明列表中的另一个工作项。
 - 始终以随机方式声明人员任务。
 - 减少该任务的潜在所有者数，例如，将该任务分配给成员较少的组。
 - 通过对用于检索列表的查询指定阈值，限制任务列表的大小。并且，请考虑使用过滤功能来限制匹配项数。可以根据任务属性进行过滤，例如，仅显示优先级为 1 的任务或者从现在开始 24 小时内到期的任务。对于直接插入任务，还可以使用定制属性或查询属性根据与该任务相关联的业务数据进行过滤。要执行此类过滤，必须在用于检索任务列表的查询中指定适当的 `WHERE` 子句。
 - 最大程度地减少或避免执行动态人员查询（即使用了替换变量的查询）。
 - 使用客户机高速缓存机制来执行人员任务查询，以避免同时运行多个查询。

缩短查询响应时间

缩短数据库响应查询时耗用的时间。

使用定制客户机时，请确保查询设置了阈值。从可用性观点看，检索数以百计或数以千计的项通常不是好的做法，这将导致执行更大量的数据库操作并延长任务的完成时间，而且用户每次只能管理少量的结果。通过指定阈值，可以最大程度地降低数据库负载和网络流量，并有助于确保客户机能够快速提供数据。

处理返回大量项的查询有一种更好的方法，即重新编写该查询，以缩小所返回的项目结果集。要实现此目标，可以仅查询某个流程实例的工作项，也可以仅查询某个日期的工作项。

还可以通过使用过滤器条件来减少查询结果。

避免扫描整个表

当使用查询应用程序编程接口（API）来列示数据库中的对象时，可以指定过滤器以缩小所检索结果的范围。在这些过滤器中，可以指定对象属性的值和范围。

处理数据库查询时，将把过滤器信息转换为结构化查询语言（SQL）语句中的 WHERE 子句。这些 WHERE 子句将对象属性映射到受影响数据库表中的列名。

如果查询指定了未转换为带索引表列的过滤器，那么该 SQL 语句可能会导致对整个表执行扫描。此扫描会对性能产生负面影响并增大发生死锁的风险。虽然可以容忍这种性能影响每天只发生几次的情况，但如果每分钟发生几次，那么会对您的工作效率产生不利影响。

在这些情况下，定制索引能够显著减轻此类影响。在一个真实的客户环境中，定制索引帮助将 API 响应时间从 25 秒缩短到 300 毫秒。并且，只需读取 6 行数据，而不必读取数据库表中全部的 724,000 行。如果对业务流程编排器数据库中的实例表定义易失标志，那么可以帮助 DB2 优化器确定适当的数据访问方案。此标志指定始终使用索引来代替扫描表，即使对于空表或几乎空的表亦如此。

根据所指定过滤器条件的不同，某些列可能未包括在索引中。如果发生这种情况，并且要执行表扫描，从而导致查询性能不佳，那么请检查语句的访问路径，例如使用 DB2Explain 执行此项检查。必要时，可以定义新索引。

优化任务查询和流程查询

用于检索任务和流程列表的 query 和 queryAll API 调用将生成对多个数据库表进行组合使用的复杂 SQL 查询。采用优化的方式表示数据有助于提高性能，对于允许多个用户同时访问任务列表的人员工作流程应用程序而言尤其如此。

关于此任务

在规模足够大的系统上，针对查询调整业务流程编排器后，响应时间通常不到一秒，即使在高负载情况下亦如此。您可以通过标准数据库计算公式来计算查询的响应时间。

对于高负载人员工作流程方案而言，最好通过查询表进行调整。查询表提供了特定查询的一组预先计算的相关数据。例如，运行查询时，数据库必须将查询属性与任务或流程实例相连接。如果使用了查询表，那么执行查询时就不再需要计算这些 SQL 连接。

实现和维护查询表要比标准数据库调整技术耗费更多的气力。因此，在使用查询表前，您应该仔细考虑标准数据库优化技术，例如索引、日志文件分布和内存。

系统支持通过两种方式实现查询表，即使用具体化视图和定制表。请根据维护成本、开发成本以及对任务和流程列表查询所返回数据的及时性要求确定是使用具体化视图还是使用定制表：


- 通过使用具体化视图来利用异步更新机制，这有助于提高查询和流程导航的性能。
 - 仅当具体化视图被使用时才进行更新
 - 设置、使用和维护相对简单
 - 可以在不更改应用程序源代码的情况下实现

- 通过使用定制表，可以将来自其他应用程序的数据包括在使用 query 或 queryAll 接口的标准查询中。此外，定制表还可用于提供任务和流程查询所需数据的优化表示。
 - 可以使用数据库触发器或其他技术以同步方式更新针对任务和流程列表查询进行了优化的定制表
 - 必须更改查询，以便查询定制表中提供的数据

相关信息

 业务流程编排器的 query() 和 queryAll 方法：最佳做法

 调整人员 workflow

 DB2 信息中心：具体化查询表

第 22 章 调整业务流程编排器资源管理器

下列建议提供了各种旨在提高业务流程编排器资源管理器性能的方法。

过程

1. 考虑增大服务器的最大堆大小。

Web 客户机必然会增加系统上的负载。连接到服务器的客户机越多，必须保存在内存中的对象就越多。因此，请考虑增大服务器的最大堆大小。这将缩短应用程序的响应时间并增大能够并行使用应用程序的最大用户数。

2. 调整 Web 容器线程池。

线程池的大小和线程不活动超时都将影响 Web 容器的性能。您可以使用管理控制台来更改这些设置（服务器 → 应用程序服务器 → **server_name** → 线程池 → **WebContainer**）。

- a. 调整最大池大小和最小池大小。

Web 客户机应用程序的所有 HTTP 请求都使用 Web 容器线程池中的线程进行处理。您可以通过调整最小池大小和最大池大小来影响 Web 客户机的性能。

池中的最大线程数并不代表应用程序服务器可以并发处理的请求数。如果池中所有线程都在使用中，那么附加的请求将进行排队，直到它们被分配到某个线程为止。如果客户机请求等待分配线程，那么客户机的响应时间将延长。但是，如果将最大数目设置得太大，系统可能会发生超负荷情况，从而导致客户机的响应时间更加长。这还可能导致其他应用程序的速度明显减慢。

要确定能否通过更改容器大小来提高性能，可以使用 Tivoli® 性能查看器来监视线程的负载（PercentMaxed 计数器）以及 Web 容器模块的活动线程数（ActiveThreads 计数器）。如果 PercentMaxed 计数器值一直是两位数，那么 Web 容器可能是瓶颈。在这种情况下，请增大线程数。如果活动线程数小于池中的线程数，那么减小线程池大小可以提高性能。

- b. 调整线程不活动超时。

线程不活动超时定义回收线程前经过的不活动时间长度，以毫秒计。如果更改此值，那么还可能会影响响应时间。值 0 表示不等待。

3. 减小大型列表的阈值。

如果正在处理大型任务列表或流程列表，那么可以降低列表的搜索限制，以避免收集用户不访问的数据。

相关任务

第 188 页的『使用管理控制台配置业务流程编排器资源管理器』
可以使用管理控制台来配置业务流程编排器资源管理器。

相关参考

第 189 页的『使用 clientconfig.jacl 脚本文件配置业务流程编排器资源管理器』
此脚本文件在服务器或集群中配置业务流程编排器资源管理器以及所有必需的资源。

第 23 章 调整业务流程编排器观察器

根据许多因素的不同，生成报告所需的时间也会有所变化。下列建议提供了各种有助于提高业务流程编排器观察器性能的方法。

选择适当的数据库管理系统

业务流程编排器观察器可以使用 DB2、Oracle 或 Derby 数据库。对于性能不是主要考虑因素并且数据量不大的开发、演示和设计原型工作而言，Derby 数据库十分理想。但是，对于生产系统而言，请使用 DB2 或 Oracle 数据库，这是因为这两个数据库系统在数据量较大时具有速度优势。

更新数据库统计信息

对于 DB2 和 Oracle 数据库而言，如果有已填充了数据的生产数据库，那么可以通过更新数据库统计信息显著提高性能。

- 要更新 DB2 数据库的统计信息，请输入下列命令：

```
RUNSTATS ON TABLE schema_prefix.EVENT_ACT_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.EVENT_PRC_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.INST_ACT_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.INST_PRC_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.OPEN_EVENTS_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.QUERY_T FOR INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.SLICES_T FOR INDEXES ALL;
```

- 要更新 Oracle 数据库的统计信息，请输入下列命令：

```
ANALYZE TABLE schema_prefix.EVENT_ACT_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.EVENT_PRC_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.INST_ACT_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.INST_PRC_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.OPEN_EVENTS_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.QUERY_T COMPUTE STATISTICS;  
ANALYZE TABLE schema_prefix.SLICES_T COMPUTE STATISTICS;
```

其中，*schema_prefix* 是创建业务流程编排器观察器数据库时使用的数据库模式的名称。有关更新数据库统计信息的更多信息，请参阅数据库文档。

仅发出相关事件

在 WebSphere Integration Developer 中，可以非常详细地定义活动或流程的日志记录。仅当还为包含活动的流程生成了活动审计事件时，业务流程编排器观察器才能识别那些事件。事件收集器应用程序将忽略无法与流程相关联的活动事件，因此不会将这些事件存储在数据库中。为了减少所发出的事件数，请执行下列步骤：

1. 选择所要审计的流程模板，并禁止发出您不感兴趣的流程的事件。
2. 从此流程模板中选择要进行审计的活动。检查是否可以在不影响报告结果的情况下省略一些事件。

为了让业务流程编排器观察器提供准确的活动或流程信息，您应该审计所有事件类型或者不审计任何事件类型。

选择用户定义函数（UDF）实现

要创建报告，业务流程编排器观察器要求在数据库中安装一些特定的 UDF。UDF 作为基于 SQL 的实现和基于 Java 的实现提供。SQL 实现的执行速度比 Java 实现快，但有一些缺点。如果您正在使用 Java 实现，那么请考虑切换到 SQL 实现。

使用独立的数据库

如果业务流程编排器观察器与业务流程编排器使用同一个数据库，那么会对它们的性能产生负面影响。如果业务流程编排器观察器使用独立的数据库，那么可以获得较好的性能。并且，请考虑在独立的数据库服务器上管理观察器数据库。

使用独立的机器

如果业务流程编排器观察器安装在包含其他应用程序（例如 BPEContainer 或 TaskContainer 应用程序）的机器上，那么请考虑在资源足以满足预期性能的独立机器上运行业务流程编排器观察器。

增大超时值

生成报告可能耗时较长。如果耗时过长，那么可能会发生事务超时或 JDBC 驱动程序连接超时。如果发生这种情况，那么请增大超时值，如下所示：

1. 在管理控制台中，转到**服务器** → **应用程序服务器** → **server_name** → **事务服务**。
2. 如果**事务生存期总超时值**小于**最大事务超时值**，那么将这两个值设置为相同。
3. 如果仍存在性能问题，那么请将**事务生存期总超时值**设置为 0 并增大**最大事务超时值**。
4. 如果仍存在性能问题，那么请将**事务生存期总超时值**和**最大事务超时值**都设置为 0，并增大 JDBC 驱动程序的连接超时值。要完成此任务，请转到数据源的连接池属性。单击 **JDBC** → **JDBC 提供程序** > **JDBC provider** → **数据源** → **data_source_name** → **连接池属性**，然后增大**连接超时值**。

在服务器集群中，必须调整所有集群成员的事务超时值。

删除不必要的数据库

报告性能取决于观察器数据库中的实例数和事件数据量。如果报告需要查询大量数据，那么性能将下降。通过减少业务流程编排器观察器数据库中的流程实例数和活动实例数，可以提高报告性能。定期删除不必要信息或旧信息有助于提高性能。

相关任务

第 218 页的『在 Java 用户定义函数与 SQL 用户定义函数之间进行选择』

通过使用 setupEventCollector 工具或运行脚本，可以对业务流程编排器观察器数据库在基于 Java 的用户定义函数（UDF）与基于 SQL 的 UDF 之间进行切换。

第 273 页的『从观察器数据库中删除数据』

通过使用管理命令，可以从业务流程编排器观察器数据库中有选择地删除所有与所指定条件匹配的流程实例的数据。通过删除不必要的数据库，可以提高生成报告时的性能。

第 7 部分 故障诊断

第 24 章 业务流程编排器配置故障诊断

使用本主题来解决与业务流程编排器及其业务流程管理器组件和人员任务管理器组件的配置相关的问题。

关于此任务

本节旨在帮助您了解业务流程管理器配置或人员任务管理器配置未按预期方式工作的原因并帮助您解决问题。下列任务侧重于确定问题以及查找配置期间可能发生的问题的解决方案。

相关信息



WebSphere Process Server 的故障诊断指南

业务流程编排器日志文件

本节描述业务流程编排器配置的日志文件位置。

概要文件的创建

业务流程编排器的概要文件操作将写概要文件工具的 logs 目录中的 `bpcaugment.log` 文件。您可以在同一目录中的 `createBPCObjects.traceout` 文件中找到更详细的跟踪信息。在 Windows 系统上，这些文件在 `install_root/logs/manageprofiles/profileName/logs` 目录中；在 Linux、UNIX 和 i5/OS 系统上，这些文件在 `install_root/logs/manageprofiles/profileName/logs` 目录中。

如果在概要文件向导中选择了样本配置选项，那么将调用 `bpeconfig.jacl` 脚本并将操作记录在概要文件 logs 目录中的 `bpeconfig.log` 文件中。此目录在 `profile_root` 目录中。

管理脚本

所有使用 `wsadmin` 运行的业务流程编排器脚本都在概要文件工具的 logs 目录中的 `wsadmin.traceout` 文件中进行记录。但是，由于每次调用 `wsadmin` 时都会覆盖此文件，因此，每次调用 `wsadmin` 前务必保存此日志文件。

与配置相关的脚本

脚本文件 `bpeconfig.jacl`、`bpeupgrade.jacl`、`clientconfig.jacl` 和 `bpeunconfig.jacl` 会将它们的日志文件写入 `logs` 目录，并且名称分别为 `bpeconfig.log`、`bpeupgrade.log`、`clientconfig.log` 和 `bpeunconfig.log`。

下列配置脚本将它们的日志分别写入 logs 目录中的 `setupObserver.log` 和 `setupEventCollector.log` 文件。

- `setUpEventCollector.bat` 和 `setUpObserver.bat` (Windows 系统)
- `setUpEventCollector.sh` 和 `setUpObserver.sh` (Linux 和 UNIX 系统)
- `setUpEventCollector` 和 `setUpObserver` (i5/OS 系统)

另外，请检查 `wsadmin.traceout` 文件。

管理实用程序脚本

ProcessChoreographer 目录中 admin 子目录中的管理脚本不写自己的日志文件。请检查 wsadmin.traceout 文件和应用程序服务器日志文件。

业务流程编排器数据库和数据源故障诊断

通过本任务解决业务流程编排器数据库和数据源的问题。

关于此任务

业务流程管理器 and 人员任务管理器都需要使用数据库。如果没有数据库，那么包含业务流程和人员任务的企业应用程序将无法工作。

- 如果正在使用 DB2:
 - 如果使用 DB2 通用 4 类 JDBC 驱动程序，并且对业务流程编排器数据源测试连接时或者在服务器启动时发生了 DB2 内部错误，例如 "com.ibm.db2.jcc.a.re:XAER_RMERR : The DDM parameter value is not supported. DDM parameter code point having unsupported value : 0x113f DB2ConnectionCorrelator: NF000001.PA0C.051117223022"，那么请执行下列操作：
 1. 检查数据源的类路径设置。在缺省设置中，WebSphere 变量 `${DB2UNIVERSAL_JDBC_DRIVER_PATH}` 可能指向 universalDriver_wbi 目录中的 WebSphere Process Server 嵌入式 DB2 通用 JDBC 驱动程序。
 2. 此驱动程序的版本可能与 DB2 服务器版本不兼容。请确保使用数据库安装版本中的原始 db2jcc.jar 文件，而不要使用 WebSphere Process Server 嵌入式 DB2 通用 JDBC 驱动程序。根据需要，将 WebSphere 变量 `${DB2UNIVERSAL_JDBC_DRIVER_PATH}` 的值更改为指向原始 db2jcc.jar 文件。
 3. 重新启动服务器。
 - 如果 DB2 实例的 db2diag.log 文件包含如下所示的 ADM5503E 消息：

```
2004-06-25-15.53.42.078000 实例: DB2 节点: 000
PID:2352(db2syscs.exe) TID:4360 Appid:*LOCAL.DB2.027785142343
数据管理 sqldEscalateLocks 探针: 4 数据库: BPEDB

ADM5503E  将表"GRAALFS.ACTIVITY_INSTANCE_T"上的"10"个锁定升级为锁定意图"X"时失败。SQLCODE 是"-911"
```

那么减小 LOCKLIST 值。例如，要将值设置为 500，请输入以下 DB2 命令：

```
db2 UPDATE DB CFG FOR BPEDB USING LOCKLIST 500
```

这将显著提高性能。
 - 为了避免死锁，请确保已将数据库系统配置为使用足够的内存，对于缓冲池而言尤其如此。对于 DB2，请使用 DB2 配置顾问程序来确定适用于配置的合理值。
 - 如果发生了与数据源实现类 `COM.ibm.db2.jdbc.DB2XADataSource` 相关的错误，那么：
 - 检查 JDBC 提供程序的类路径定义是否正确。
 - 检查组件管理的认证别名是否设置为 `BPCDB_nodeName.serverName_Auth_Alias`（如果业务流程编排器被配置在服务器上）或 `BPCDB_clusterName_Auth_Alias`（如果业务流程编排器被配置在集群中）。
- 如果使用 Derby:

- 如果在 Linux 或 UNIX 系统上发生了“打开的文件过多”错误，那么增大可用文件句柄数，例如，增大到 4000 或更大。有关如何增大可用文件句柄数的更多信息，请参阅操作系统的文档。
- 如果尝试调用 ij 命令行处理器时发生“找不到 Java 类”异常，那么请确保已对 Java 环境进行设置并且 classpath 环境变量包含下列 JAR 文件：
 - derby.jar
 - derbytools.jar
- 如果正在使用嵌入式 Derby 驱动程序，并且无法使用 Derby 工具（例如 ij）连接至 Derby 数据库，那么将发生以下异常：

ERROR XJ040: 未能启动数据库"c:\WebSphere\AppServer\profiles\profile_name\databases\BPEDB",
 请参阅下一个异常以了解详细信息。
 ERROR XSDB6: 另一个 Derby 实例可能已启动数据库
 c:\WebSphere\AppServer\profiles\profile_name\databases\BPEDB。

请确保每次只有一个应用程序访问 Derby 数据库。

- 如果您编写的客户机使用了业务流程编排器 API，但未首先认证用户，那么应该修改该客户机，以便在使用 API 前执行登录。执行迁移后，J2EE 角色 BPEAPIUser 和 TaskAPIUser 将设置为值 Everyone，这通过保持 6.0.x 行为（即使启用了应用程序安全性也不要要求进行登录）维护向后兼容性。修正客户机后，必须将这些角色更改为值 AllAuthenticated，以防未经认证的用户访问 API。对于新安装版本而言，这些角色将具有缺省值 AllAuthenticated。
- 如果安装包含业务流程或人员任务的企业应用程序时发生数据库错误，那么请确保业务流程容器使用的数据库系统处于运行状态并且可访问。安装企业应用程序时，会将所有流程模板和任务模板写入业务流程编排器数据库。
- 使用本地字符时发生了问题。请确保创建的数据库支持 Unicode 字符集。
- 如果在数据库中找到表和视图，并且未启用“创建模式”选项，那么请检查下列各项：
 - 如果已配置数据库模式限定符，那么请检查下列各项：
 - 模式限定符必须与数据库中的模式匹配。它必须是脚本中使用的模式。
 - 必须授权该用户使用那些数据库表和视图。
 - 如果未配置模式限定符，那么请确保：
 - 该用户的认证别名必须是用于运行脚本的用户的标识，或者必须与脚本中使用的模式限定符匹配。
 - 必须授权该用户使用那些数据库表和视图。
- 如果已启用“创建模式”选项，并且找不到数据库表和视图，那么将按如下方式自动创建数据库表和对象：
 - 如果已配置模式限定符，那么将使用该模式限定符来创建表和视图。
 - 如果未配置模式限定符，那么将使用该用户标识来创建表和视图。

启用替代功能后，任务容器无法启动

您已创建新的业务流程编排器配置并启用替代功能，或者已对现有配置启用替代功能。

症状

TaskContainer_*suffix* 应用程序无法启动。

原因

已启用人员替代功能，但未正确地对其进行配置。

解决方案

要使替代功能生效，请执行下列操作：

- 必须启用应用程序安全性。
- 人员任务管理器必须使用虚拟成员管理器（VMM）人员目录提供程序（联合存储库）。
- 如果要使用 LDAP 目录，那么必须将 VMM 配置为使用 LDAP。

作为临时变通方法，可以对人员任务管理器禁用替代功能，然后重新启动任务容器：

1. 在管理控制台中，单击**服务器** → **应用程序服务器** → **server_name** 或 **服务器** → **集群** → **cluster_name**，接着在**业务集成**中展开**业务流程编排器**，然后单击**人员任务管理器** → **运行时**
2. 清除**启用替代**选项。
3. 单击**应用**。
4. 重新启动 TaskContainer_*suffix* 应用程序。

再次启用替代功能前，请确保执行下列操作：

1. 第 106 页的『规划人员目录提供程序』
2. 第 173 页的『配置虚拟成员管理器人员目录提供程序』
3. 第 180 页的『配置人员替代功能』

相关任务

第 580 页的『人员分配故障诊断』

使用以下信息来帮助解决与人员授权角色分配相关的问题。

6.0.x 业务流程编排器 API 客户机在 6.1 环境中发生故障

您在升级到 WebSphere Process Server V6.1 时未迁移 6.0.x 业务流程编排器 API 客户机。当您尝试在 6.1 环境中运行该客户机时，该客户机发生故障。

症状

SystemOut.log 文件包含类似于以下的异常：

```
[9/6/07 21:05:27:093 PDT] 00000045 ExceptionUtil E CNTR0020E: EJB threw an unexpected (non-declared) exception during invocation of method "processMessage" on bean "BeanId(validateDataApp#validateDataEJB.jar#component.validateItem, null)".  
Exception data: javax.ejb.AccessLocalException: ;  
nested exception is: com.ibm.websphere.csi.CSIAccessException:  
SECJ0053E: Authorization failed for /UNAUTHENTICATED while invoking  
(Home)com/ibm/bpe/api/BusinessFlowManagerHome create:4  
securityName: /UNAUTHENTICATED;accessID: UNAUTHENTICATED is not granted any of the required  
roles: BPEAPIUser  
com.ibm.websphere.csi.CSIAccessException: SECJ0053E: Authorization failed for  
/UNAUTHENTICATED while invoking (Home)com/ibm/bpe/api/BusinessFlowManagerHome
```



```
create:4 securityName: /UNAUTHENTICATED;accessID: UNAUTHENTICATED is not granted any of the required
roles: BPEAPIUser
at com.ibm.ws.security.core.SecurityCollaborator.performAuthorization(SecurityCollaborator.java:484)
at com.ibm.ws.security.core.EJSSecurityCollaborator.preInvoke(EJSSecurityCollaborator.java:218)
at com.ibm.ejs.container.EJSContainer.preInvokeForStatelessSessionCreate(EJSContainer.java:3646)
at com.ibm.ejs.container.EJSContainer.preInvoke(EJSContainer.java:2868)
at com.ibm.bpe.api.EJSLocalStatelessGenericBusinessFlowManagerEJBHome_a412961d.create(Unknown Source)
```

原因

如果业务流程编排器 API 客户机未包含用户认证信息，那么将依赖于一个安全漏洞。在 WebSphere Process Server V6.1 中，已修正此安全漏洞，这导致该客户机发生故障。

解决方案

修改 API 客户机，强制使用户登录该客户机，然后才允许他们使用 API。

作为临时变通方法，可以更改 BPEAPIUser 和 TaskAPIUser 角色的映射。要更改映射，请执行下列操作：

1. 在管理控制台中，单击应用程序 → 企业应用程序 → **BPEContainer_suffix**，然后在详细属性中单击安全角色到用户/组的映射。
2. 将 BPEAPIUser 角色由所有已认证的用户更改为每个人，然后单击**确定**。
3. 对 TaskContainer_suffix 和 TaskAPIUser 角色重复步骤 2。
4. 修改客户机后，必须将这些角色重新更改为所有已认证的用户，以防未经认证的用户访问 API。

对业务流程编排器启用跟踪

本节描述与支持人员联系前执行的操作。

启用跟踪

业务流程编排器的跟踪功能使用标准的 WebSphere Process Server 跟踪机制。必须以正常方式启用此机制。

指定跟踪的方式如下所示：

```
com.ibm.bpe.*=all:com.ibm.task.*=all:com.ibm.ws.staffsupport.*=all
```

其中 com.ibm.bpe.*=all 跟踪业务流程，而 com.ibm.task.*=all 跟踪人员任务。人员任务的其余方面，即人员目录提供程序，由 com.ibm.ws.staffsupport 跟踪。

向支持人员发送的内容

启用跟踪后，请再现问题场景，然后向支持人员提供下列文件：

- ffdc 文件夹中的 WebSphere Application Server FFDC 日志
- 下列日志文件：
 - SystemOut.log
 - SystemErr.log
 - trace.log

在 Linux、UNIX 和 i5/OS 系统上，这些文件在 *profile_root/logs/server_name* 目录中。在 Windows 平台上，这些文件在 *profile_root\logs\server_name* 目录中。

如果问题场景导致记录了大量内容，那么可能使用 SystemOut_07.10.01_11.00.51.log 之类的名称创建了日志的备份文件。您可以使用管理控制台来更改所创建的备份文件数以及日志文件的大小。最好同时增大这两个值，以确保捕获所有数据。

相关信息

 [WebSphere Process Server 的故障诊断指南](#)

第 25 章 业务流程和人员任务故障诊断

使用本主题来解决与业务流程和人员任务相关的问题。

关于此任务

下列任务侧重于解决业务流程或任务执行期间发生的问题。

对业务流程应用程序和人员任务应用程序的安装进行故障诊断

在 ND 环境中安装包含业务流程和/或人员任务的应用程序时，Deployment Manager 的 SystemErr.log 文件记录了异常。

症状

在 ND 环境中安装包含业务流程和/或人员任务的应用程序时，Deployment Manager 的 SystemErr.log 文件记录了以下异常：

```
SystemErr R com.ibm.ws.management.commands.sib.SIBAdminCommandException:
CWSJA0012E: Messaging engine not found.
at com.ibm.ws.management.commands.sib.SIBAdminCommandHelper.createDestination
(SIBAdminCommandHelper.java:787)
at com.ibm.ws.management.commands.sib.CreateSIBDestinationCommand.afterStepsExecuted
(CreateSIBDestinationCommand.java:459)
at com.ibm.websphere.management.cmdframework.provider.AbstractTaskCommand.execute
(AbstractTaskCommand.java:547)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdminHelper.call(SIBAdminHelper.java:136)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdminHelper.createSIBDestination
(SIBAdminHelper.java:112)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdmin.createDestination(SIBAdmin.java:327)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.createDestination
(SIBDestinationTask.java:263)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.preInstallModule
(SIBDestinationTask.java:71)
at com.ibm.ws.sca.internal.deployment.SCATaskBase.installModule(SCATaskBase.java:57)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.processArtifacts
(SIBDestinationTask.java:228)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.install
(SIBDestinationTask.java:287)
at com.ibm.ws.sca.internal.deployment.SCAInstallTask.performInstallTasks
(SCAInstallTask.java:116)
at com.ibm.ws.sca.internal.deployment.SCAInstallTask.performTask
(SCAInstallTask.java:61)
at com.ibm.ws.management.application.SchedulerImpl.run(SchedulerImpl.java:253)
at java.lang.Thread.run(Thread.java:568)
```

原因

缺少“SCA.SYSTEM.cellName.Bus”总线的总线成员。

解决方案

在管理控制台中，单击**服务集成** → **总线** → **SCA.SYSTEM.cellName.Bus**。在“拓扑”部分中，单击**总线成员**。添加要安装业务流程或人员任务应用程序的服务器或集群作为总线成员，然后重新启动受影响的服务器或集群并再次尝试安装该应用程序。

对业务流程的执行进行故障诊断

本节描述常见业务流程执行问题的解决方案。

关于此任务

在业务流程编排器资源管理器中，可以通过 IBM 技术支持页面搜索错误消息代码。

过程

1. 在错误页面上，单击[搜索更多信息](#)链接。这将开始在 IBM 技术支持站点上搜索错误代码。此站点仅提供英文版资料。
2. 将错误页面中显示的错误消息代码复制到剪贴板。错误代码的格式为 CWWBcnnnnc，其中每个 c 是一个字符，nnnn 是一个 4 位数。转到 WebSphere Process Server 技术支持页面。
3. 将该错误代码粘贴到 **Additional search terms** 字段中，然后单击 **Go**。

下一步做什么？

下列主题提供了特定问题的解决方案。

停止包含微流程的应用程序时发生 **ClassCastException**

SystemOut.log 文件中记录了包含微流程的应用程序在停止期间发生的 **ClassCastException** 异常。

原因

停止应用程序时，将从类路径中除去 EAR 文件中包含的类。但是，需要这些类的微流程实例可能仍在执行中。

解决方案

请执行下列操作：

1. 首先停止该微流程的流程模板。从此刻起，不可能再根据该模板启动新的微流程实例。
2. 至少等待微流程的最长执行持续时间，以使所有运行中的实例有时间完成。
3. 停止该应用程序。

调用 **processMessage** 方法期间发生意外的异常（消息：**CNTR0020E**）

业务流程容器已停止，客户机未能连接至服务器。

解决方案

验证业务流程容器是否处于运行状态。

XPath 查询从数组中返回了意外的值

使用 XPath 查询来访问数组成员时，返回了意外的值。

原因

此问题的一个常见原因是，系统假定第一个数组元素的下标值为 0。在 XPath 查询中，第一个数组元素的下标值为 1。

解决方案

检查所使用的第一个数组下标值是否为元素 1。

活动由于未处理的故障而停止（消息：CWWBE0057I）

系统日志包含 CWWBE0057I 消息，流程处于“正在运行”状态，但未沿着当前路径继续进行导航。

原因

如果下列情况全都存在，那么调用活动、直接插入人员任务和 Java 片段将处于“已停止”状态：

- 该活动引起故障
- 在外层作用域中未处理该故障
- 该活动的 `continueOnError` 属性设置为 `no`。

解决方案

此问题的解决方案要求在两个级别执行操作：

1. 管理员必须手动修复已停止的活动实例。例如，强制完成或强制再次执行已停止的活动实例。
2. 必须调查故障原因。在某些情况下，此故障是由模型中必须更正的设计错误导致的。

第 391 页的『管理业务流程的生命周期』

在调用可以启动流程的业务流程编排器 API 方法时，将创建流程实例。流程实例的导航将一直持续到它的所有活动都处于结束状态为止。您可以对流程实例执行各种操作，以便对其生命周期进行管理。

第 403 页的『修复活动』

长时间运行流程可以包含运行时间同样较长的活动。这些活动可能会遇到未捕获的错误并进入“已停止”状态。处于运行状态的活动也可能会表现为未响应。在这两种情况下，流程管理员都可以通过许多方法对该活动执行操作，以使该流程的导航能够继续进行。

未对微流程进行补偿

某个微流程调用了服务，该流程失败，但未调用撤销服务。

解决方案

要触发微流程补偿，必须符合许多条件。请检查下列各项：

1. 登录业务流程编排器资源管理器，然后单击**失败的补偿**以检查补偿服务是否已失败并需要进行修复。
2. 仅当回滚微流程的事务时，才会触发微流程补偿。请检查情况是否如此。
3. 必须将微流程的 `compensationSphere` 属性设置为 `required`。

4. 仅当相应的转发服务未参与微流程的事务时，才会运行补偿服务。请确保转发服务未参与导航事务，例如，在对流程组件进行引用时，将服务组件体系结构（SCA）限定符 `suspendTransaction` 设置为 `True`。

长时间运行流程似乎已停止

长时间运行流程处于“正在运行”状态，但它似乎未执行任何操作。

原因

此类行为的可能原因很多：

1. 导航消息的重试次数过多，因此已移入保留队列或挂起队列。
2. 服务组件体系结构（SCA）基础结构发出的应答消息反复失败。
3. 该流程正在等待某个事件、已超时或者正在等待长时间运行调用或任务返回。
4. 该流程中的某个活动处于“已停止”状态。

解决方案

对于上述每个原因，更正操作各不相同：

1. 使用管理控制台查询和重放已失败的消息。
2. 在管理控制台的事件管理视图中检查是否有任何已失败的事件。
 - 如果存在任何与服务组件体系结构（SCA）应答消息相关的失败事件，请重新激活那些消息。
 - 否则，请强制完成或强制再次执行该长时间运行活动。
3. 检查是否存在处于“已停止”状态的活动并修复这些活动。如果系统日志包含 `CWWBE0057I` 消息，那么可能还需要更正模型，如消息：`CWWBE0057I` 所述。

调用另一个 EAR 文件中的同步子流程失败

如果长时间运行流程以同步方式调用另一个流程，并且该子流程在另一个企业归档（EAR）文件中，那么子流程调用将失败。

生成的异常如下所示：

```
com.ibm.ws.sca.internal.ejb.util.EJBStubAdapter com.ibm.ws.sca.internal.ejb.util.EJBStubAdapter#003
Exception:
java.rmi.AccessException: CORBA NO_PERMISSION 0x49424307 No; nested exception is:
org.omg.CORBA.NO_PERMISSION: The WSCredential does not contain a forwardable token.
Please enable Identity Assertion for this scenario.
vmcid: 0x49424000 minor code: 307 completed: No
at com.ibm.CORBA.iiop.UtilDelegateImpl.mapSystemException(UtilDelegateImpl.java:202)
at javax.rmi.CORBA.Util.mapSystemException(Util.java:84)
```

原因

由于子流程调用将致使执行远程 EJB 方法调用，所以，如果要调用另一个 EAR 文件中的同步子流程，那么必须启用 Common Secure Interoperability V2（CSIv2）身份声明。

解决方案

配置 CSIv2 入站认证和 CSIv2 出站认证。

执行期间发生意外的异常（消息：CWWBA0010E）

队列管理器未处于运行状态，或者业务流程编排器配置包含不正确的数据库密码。

解决方案

请检查下列各项：

1. 如果 systemout.log 文件包含 "javax.jms.JMSException: MQJMS2005: failed to create MQQueueManager"，那么启动队列管理器。
2. 确保业务流程编排器配置中存储的数据库管理员密码与数据库中设置的密码匹配。

未知的事件（消息：CWWBE0037E）

在尝试向流程实例发送事件或者启动新的流程实例时，发生了“CWWBE0037E: 未知的事件”异常。

原因

此错误的一个常见原因是，向某个流程发送了消息，但由于已执行接收或检取活动，所以此流程实例无法再次使用该消息。

解决方案

要更正此问题：

- 如果该事件应该由现有流程实例处理，那么必须传递与尚未执行相应接收或检取活动的现有流程实例匹配的关联集值。
- 如果该事件将启动新的流程实例，那么关联集值不能与现有流程实例相匹配。

有关在业务流程中使用关联集的更多信息，请参阅技术说明 1171649。

找不到并且无法创建流程实例（消息：CWWBA0140E）

在尝试向流程实例发送事件时，系统发出了 'CreateRejectedException' 消息。

原因

此错误的一个常见原因是，向接收或检取活动发送了消息，但由于该活动的 createInstance 属性设置为 no，并且随此活动所使用关联集的消息一起传递的值未与任何现有流程实例相匹配，因此无法创建新的流程实例。

解决方案

要解决此问题，必须传递与现有流程实例匹配的关联集值。

有关在业务流程中使用关联集的更多信息，请参阅 BPEL 流程中的关联集。

流程实例的“已失败”状态不允许执行所请求的 sendMessage 操作（消息：CWWBE0126E）

在尝试向流程实例发送事件时，系统发出了 'EngineProcessWrongStateException' 消息。

原因

此错误的一个常见原因是，向接收或检取活动发送了用于创建新流程实例的消息，但无法将新流程实例实例化。如果随消息一起为此活动所使用的关联集传递的值与处于“已失败”状态的流程实例匹配，那么将发生这种情况。

解决方案

要解决此问题，必须删除现有流程实例或者传递与现有流程实例不匹配的关联集值。有关在业务流程中使用关联集的更多信息，请参阅 BPEL 流程中的关联集。

Java 片段包含未初始化的变量或者发生 NullPointerException

在业务流程中使用未初始化的变量会导致各种异常。

症状

如下所示的异常：

- 执行读取或处理变量内容的 Java 片段或 Java 表达式期间，抛出 NullPointerException。
- 执行分配、调用、应答或抛出活动期间，抛出 BPEL 标准故障“uninitializedVariable”，即消息 CWWBE0068E。

原因

业务流程启动时，该流程中的所有变量都具有空值，即不会预先对变量进行初始化。在 Java 片段或 Java 表达式中使用未初始化的变量将导致 NullPointerException。

解决方案

在使用变量前，必须对变量进行初始化。可以通过赋值活动来初始化变量，例如在赋值操作的 to-spec 中指定该变量，也可以在 Java 片段中初始化变量。

标准故障异常“missingReply”（消息：CWWBE0071E）

执行微流程或长时间运行流程时发生了 BPEL 标准故障“missingReply”（消息：CWWBE0071E），或者在系统日志或 SystemOut.log 文件中找到此错误。

原因

双向操作必须发送应答。如果流程在未执行应答活动的情况下结束，那么将发生此错误。在下列任何情况下，都会发生此错误：

- 跳过了应答活动。
- 发生了故障，但相应的故障处理程序未包含应答活动。
- 发生了故障，但没有相应的故障处理程序。

解决方案

更正模型，确保流程结束前始终执行应答活动。

并行路径序列化

流程活动包含两个以上的并行调用活动，但这些调用活动按顺序运行。

解决方案

- 要实现真实的并行性，每条路径都必须在独立的事务中。请将所有并行调用活动的“事务行为”属性设置为“提前落实”或“需要自身”。
- 如果使用 Derby 或 Oracle 作为数据库系统，那么流程引擎会将并行路径的执行序列化。您无法更改此行为。

将嵌套的数据对象复制到另一个数据对象将破坏源对象中的引用

数据对象 Father 包含另一个数据对象 Child。在 Java 片段或客户机应用程序中，数据对象 Mother 的子结构对包含 Child 的对象进行了访存和设置。数据对象 Father 中对 Child 的引用将消失。

原因

对 Child 的引用已从 Father 移至 Mother。

解决方案

如果要在 Java 片段或客户机应用程序中执行此类数据变换，并且要保留 Father 中的引用，那么在将该数据对象赋给另一个对象之前，请对其进行复制。以下代码段说明了如何执行此操作：

```
BOCopy copyService = (BOCopy)ServiceManager.INSTANCE.locateService
    ("com/ibm/websphere/bo/BOCopy");
DataObject Child = Father.get("Child");
DataObject BCopy = copyService.copy(Child);
Mother.set("Child", BCopy);
```

CScope 不可用

在长时间运行流程中启动微流程或运行导航步骤失败，返回的消息为“后续条件违例!(cscope != null)”。

原因

在某些情况下，流程引擎需要使用补偿服务，但该服务尚未启用。

解决方案

按照 PDF for Administering 中描述的方式启用补偿服务。

处理与流程或任务相关的消息

本节描述如何进一步了解屏幕上显示的或日志文件中包含的业务流程编排器消息。

关于此任务

业务流程编排器的相关消息带有 CWWB 前缀（表示与流程相关的消息）或 CWTK 前缀（表示与任务相关的消息）。这些消息的格式是 *PrefixComponentNumberTypeCode*。类型码可以是：

- I** 参考消息
- W** 警告消息
- E** 错误消息

当流程和任务运行时，消息将显示在业务流程编排器资源管理器中，或者将被添加到 SystemOut.log 文件和跟踪输出中。如果这些文件中提供的消息文本不足以帮助您解决问题，那么您可以使用 WebSphere Application Server 症状数据库来查找更多信息。要查看业务流程编排器消息，请使用 WebSphere 日志分析器来检查 activity.log 文件。

过程

1. 启动 WebSphere 日志分析器。

运行下列其中一个脚本：

- 在 Windows 系统上：*install_root/bin/waslogbr.bat*
- 在 Linux、UNIX 和 i5/OS 平台上：*install_root/bin/waslogbr.sh*

2. 可选：单击文件 → 更新数据库 → **WebSphere Application Server 症状数据库** 以搜索最新版本的症状数据库。
3. 可选：装入活动日志。
 - a. 选择活动日志文件。
 - 在 Windows 系统上：*profile_root\profiles\profile_name\logs\activity.log* 文件
 - 在 Linux、UNIX 和 i5/OS 系统上：*profile_root/profiles/profile_name/logs/activity.log* 文件
 - b. 单击打开。

对业务流程和人员任务的管理进行故障诊断

本文描述如何解决业务流程和人员任务的一些常见问题。

关于此任务

下列信息可以帮助您解决业务流程和人员任务的问题。

- 当业务流程应用程序仍包含流程实例时，如果您尝试将其停止，那么管理控制台将停止响应。在尝试停止该应用程序之前，必须停止业务流程以免创建任何新实例，然后执行下列其中一项操作：
 - 等待现有的所有流程实例平稳结束。
 - 终止并删除所有流程实例。只有完成这些操作后，才能安全地停止该流程应用程序。有关如何预防此问题的更多信息，请参阅技术说明 1166009。
- 当人员任务应用程序仍包含任务实例时，如果您尝试将其停止，那么管理控制台将停止响应。要停止该应用程序，您必须：
 1. 停止人员任务，以便不允许创建任何新实例。
 2. 请执行下列其中一项操作：
 - 等待现有的所有任务实例平稳结束。
 - 终止并删除所有任务实例。
 3. 停止该任务应用程序。
- 由调用任务启动的长时间运行业务流程未能启动。一个 JSP 片段使该调用任务可供用户使用。在以下示例中，使用了同步调用模式 `createAndCallTask`。在这种情况下，长时间运行业务流程将无法启动：

```

HumanTaskManager htm = ...
TaskTemplate taskTemplate = htm.getTaskTemplate( "start the process" );
Task task = htm.createAndCallTask( taskTemplate.getTKID() );
while (task.getState() != TASK.TASK_STATE_FINISHED)
{
    Sleep(100);
}

```

长时间运行流程由多个事务组成，它采用异步调用样式。因此，您必须使用异步调用模式 `createAndStartTask` 来启动此流程。

```

HumanTaskManager htm = ...
TaskTemplate taskTemplate = htm.getTaskTemplate( "start the process" );
Task task = htm.createAndStartTask( taskTemplate.getTKID() );
while (task.getState() != TASK.TASK_STATE_FINISHED)
{
    Sleep(100);
}

```

此外，必须将 JSP 部署描述符中的 `transaction` 属性设置为 `NotSupported`。这确保在不使用事务的情况下执行该代码片段，并且 `createAndStartTask` 方法在启动流程实例时将打开新事务。此事务将在 `createAndStartTask` 方法返回时落实，并且消息可视。

最好为除“已完成”以外的状态准备“while”循环。例如，在流程执行期间，如果某个活动失败，那么结束状态可能是 `TASK.TASK_STATE_FAILED`。

升级电子邮件故障诊断

使用本信息来解决与升级电子邮件相关的问题。

关于此任务

当人员任务的进度未如预期时，将触发升级。升级将创建工作项。它还可以向受升级影响的用户发送电子邮件。如果发生与升级电子邮件相关的问题，那么使用此处的信息将有助于解决这些问题。

- 在 `SystemOut.log` 文件中查找与人员分配或电子邮件地址相关的错误消息。
- 如果 `SystemOut.log` 文件未包含任何相关消息，那么请对邮件会话服务器启用调试方式。

在管理控制台中，单击 **资源** → **邮件** → **邮件会话** → **HTMMailSession_server**，然后选中 **启用调试方式** 复选框。发送升级电子邮件时，调试信息将被写入 `SystemOut.log` 文件。

- 如果正在使用虚拟成员管理器作为人员目录提供程序，并且发生与电子邮件地址相关的问题，那么请启用 `Staff.Diagnosis` 定制属性。
 1. 在管理控制台中，单击 **应用程序服务器** → **server_name**。
 2. 在 **业务集成** 中，展开 **业务流程编排器**，然后单击 **人员任务管理器配置**。
 3. 在 **配置选项卡** 的 **其他属性** 中，单击 **定制属性** → **Staff.Diagnosis**，然后在 **值** 字段中输入 `on`。

发送升级电子邮件时，有关人员分配的其他信息将被写入 `SystemOut.log` 文件。

- 检查人员任务管理器挂起队列是否包含消息。
 1. 在管理控制台中，单击 **应用程序服务器** → **server_name**。
 2. 在 **业务集成** 中，展开 **业务流程编排器**，然后单击 **人员任务管理器配置**。

3. 在运行时选项卡中，单击**重放挂起队列**。挂起队列中的消息将显示在挂起队列消息字段中。

如果挂起队列包含消息，那么请检查服务器的第一次故障数据捕获（FFDC）目录，以了解更多有关错误的信息。

- 检查与电子邮件重新发送次数和电子邮件发送超时相关的定制属性值。
 1. 在管理控制台中，单击**应用程序服务器** → **server_name**。
 2. 在**业务集成**中，展开**业务流程编排器**，然后单击**人员任务管理器配置**。
 3. 在**配置选项卡**的**其他属性**中，单击**定制属性**。
 4. 检查 **EscalationEmail.RetryTimeout** 和 **EscalationEmail.MaxRetries** 字段的值。

EscalationEmail.RetryTimeout

指定人员任务管理器在重新发送失败的电子邮件通知前等待的时间长度。此字段的缺省值是 3600 秒（1 小时）。每次重试失败时，重试超时都将动态地加倍延长。缺省情况下，如果第一次重试失败，那么下一次重试将在两小时后进行。

EscalationEmail.MaxRetries

指定人员任务管理器尝试重新发送失败的电子邮件通知的次数。此字段的缺省值是 4。如果将此字段的值设置为 0，那么不会重新发送失败的电子邮件通知。如果所有重试都失败，那么将把一条消息写入挂起队列。在管理控制台的**运行时选项卡**中，可以查看挂起队列中人员任务管理器的消息。重放消息相当于第一次发送电子邮件。

人员分配故障诊断

使用以下信息来帮助解决与人员授权角色分配相关的问题。

关于此任务

本信息阐述下列问题：

- 部署人员目录提供程序期间出错
- 工作项分配未反映人员目录中的条目
- 工作项分配未立即反映对人员目录所作的更改
- 任务或流程实例的意外人员分配
- 人员任务停止
- 与人员分配相关的错误和警告消息
- 与组工作项和“Group”人员分配条件相关的问题
- 清除已存储的人员分配结果

您还可以在技术支持搜索页面中搜索其他信息。

部署人员目录提供程序期间出错

如果正在使用轻量级目录访问协议（LDAP）人员目录提供程序，那么部署可能会由于提供程序配置参数值不正确而失败。

- 请确保设置所有必需参数。

- 要将 baseDN 参数设置为 LDAP 目录树的根，请指定空字符串；请将 baseDN 参数设置为两个撇号 (') 字符 (')。请不要使用双引号 (")。设置 baseDN 参数失败将导致部署时发生 NullPointerException 异常。

工作项分配未反映人员目录中的条目

人员查询所检索的最大用户标识数由 Threshold 变量指定，此变量是在所使用的 XSL 变换文件中定义的。例如，LDAP 人员目录提供程序使用的 XSL 变换文件是 LDAPTransformation.xsl。在 Linux、UNIX 和 i5/OS 平台上，此文件在 *install-root/ProcessChoreographer/Staff* 目录中；在 Windows 平台上，此文件在 *install-root\ProcessChoreographer\Staff* 目录中。缺省 Threshold 值是 20。要更改此值，请执行下列操作：

1. 创建新的人员目录提供程序配置，并提供您自己的 XSL 文件版本。
2. 根据需要，修改 XSL 文件中的以下条目：

```
<xsl:variable name="Threshold">20</xsl:variable>
```

注：如果指定较大的 Threshold 值，那么会导致性能下降。因此，请不要指定大于 100 的值。

工作项分配未立即反映对人员目录所作的更改

业务流程编排器将根据 LDAP 服务器等人员目录进行人员分配评估的结果高速缓存在运行时数据库中。人员目录中的内容发生更改时，数据库高速缓存并不会立即反映这些更改。

《管理指南》描述了三种刷新此高速缓存的方法：

- **使用管理控制台刷新人员查询结果。**如果进行了重大更改并且需要刷新几乎所有人员查询的结果，那么使用此方法。
- **使用管理命令刷新人员查询结果。**如果您使用 wsadmin 工具编写了管理脚本，或者要立即刷新一小部分人员查询结果，那么使用此方法。
- **使用刷新守护程序刷新人员查询结果。**如果要设置对所有已到期的人员查询结果进行定期自动刷新，那么使用本方法。

注：一个用户登录后，如果该用户的组成员资格发生更改，那么上述方法均无法刷新 Group 所返回的高速缓存值。组成员资格信息高速缓存在用户的登录会话（WebSphere 安全性 LTPA 令牌）中，而缺省情况下此会话将在两小时后到期。用于进行流程导航的流程启动者标识的组成员资格列表永远不会被刷新。

任务或流程实例的意外人员分配

如果没有为任务的某些角色定义人员分配条件，或者人员分配失败或未返回结果，那么将以缺省方式进行人员分配。这些缺省值可能会导致进行用户授权时出现意外情况；例如，流程启动者获得了流程管理员权限。此外，许多权限将由从属工件继承。例如，流程管理员还可能成为所有直接插入任务的管理员。

下表举例说明了各种情况下应用的缺省值:

表 91. 业务流程的角色

业务流程的角色	在流程模型中未定义此角色的情况下 ...	在流程模型中定义了此角色, 但人员分配失败或者未返回正确结果的情况下 ...
流程管理员	流程启动者成为流程管理员	发生以下异常, 并且该流程无法启动: EngineAdministratorCannotBeResolvedException
流程阅读者	没有阅读者	没有阅读者

表 92. 直接插入人员任务及其升级的角色

直接插入人员任务及其升级的角色	在任务模型中未定义此角色的情况下 ...	在任务模型中定义了此角色, 但人员分配失败或者未返回正确结果的情况下 ...
任务管理员	仅应用继承规则	仅应用继承规则
任务潜在实例创建者	所有用户都成为潜在实例创建者	所有用户都成为潜在实例创建者
任务潜在启动者	所有用户都成为潜在启动者	所有用户都成为潜在启动者
任务潜在所有者	所有用户都成为潜在所有者	管理员成为潜在所有者
任务编辑者	没有编辑者	没有编辑者
任务阅读者	仅应用继承规则	仅应用继承规则
升级接收者	管理员成为升级接收者	管理员成为升级接收者

对于直接插入任务, 将应用下列继承规则:

- 流程管理员成为所有直接插入任务、它们的子任务、后续任务和升级的管理员。
- 流程阅读者成为所有直接插入任务、它们的子任务、后续任务和升级的阅读者。
- 任务管理员成为所有子任务、后续任务以及所有这些任务的升级的管理员。
- 任务阅读者成为所有子任务、后续任务以及所有这些任务的升级的阅读者。
- 任何任务角色的成员都成为此任务的升级、子任务和后续任务的阅读者。
- 升级接收者成为已升级任务的阅读者。

表 93. 独立人员任务及其升级的角色

独立人员任务及其升级的角色	在任务模型中未定义此角色的情况下 ...	在任务模型中定义了此角色, 但人员分配失败或者未返回正确结果的情况下 ...
任务管理员	发起者成为管理员	任务无法启动
任务潜在实例创建者	所有用户都成为潜在实例创建者	所有用户都成为潜在实例创建者
任务潜在启动者	发起者成为潜在启动者	任务无法启动
潜在所有者	所有用户都成为潜在所有者	管理员成为潜在所有者
编辑者	没有编辑者	没有编辑者
阅读者	仅应用继承规则	仅应用继承规则

表 93. 独立人员任务及其升级的角色 (续)

独立人员任务及其升级的角色	在任务模型中未定义此角色的情况下 ...	在任务模型中定义了此角色, 但人员分配失败或者未返回正确结果的情况下 ...
升级接收者	管理员成为升级接收者	管理员成为升级接收者

对于独立任务, 将应用下列继承规则:

- 任务管理员成为所有子任务、后续任务以及所有这些任务的升级的管理员。
- 任务阅读者成为所有子任务、后续任务以及所有这些任务的升级的阅读者。
- 任何任务角色的成员都成为此任务的升级、子任务和后续任务的阅读者。
- 升级接收者成为已升级任务的阅读者。

注: 使用业务流程管理器 API 来调用方法时, BPESystemAdministrator 角色的成员具有管理员权限, BPESystemMonitor 角色的成员具有阅读者权限。

注: 使用人员任务管理器 API 来调用方法时, TaskSystemAdministrator 角色的成员具有管理员权限, TaskSystemMonitor 角色的成员具有阅读者权限。

人员任务停止

如果遇到下列一个或多个问题:

- 无法声明人员任务, 即使业务流程已成功地开始进行导航。
- SystemOut.log 文件包含以下消息: CWWB0057I: 由于发生了未处理的故障, 流程"MyProcess"的活动"MyStaffActivity"已停止...

这些问题表明未启用 WebSphere Application Server 安全性。使用人员授权的人员任务和流程要求启用安全性并配置用户注册表。请执行下列步骤:

1. 检查是否已启用 WebSphere 安全性。在管理控制台中, 转至安全性 → 全局安全性并确保启用全局安全性复选框已选中。
2. 检查是否已配置用户注册表。在管理控制台中, 转至安全性 → 用户注册表并检查活动用户注册表属性。
3. 如果活动已停止, 那么将其重新启动。

与人员分配相关的错误和警告消息

在人员分配期间访问人员目录时, 可能会发生一些常见错误。要查看这些错误的详细信息, 可以使用下列跟踪设置来启用跟踪: com.ibm.bpe.*=all: com.ibm.task.*=all:com.ibm.ws.staffsupport.ws.*=all。

警告或错误消息将指示下列常见错误情况:

- trace.log 文件中的未能连接至 LDAP 服务器表明未能连接至 LDAP 服务器。请检查网络设置, 检查所使用的人员目录提供程序的配置 (尤其是提供程序 URL), 并验证 LDAP 服务器是否需要 SSL 连接。
- System.out 或 System.err 文件中的 javax.xml.transform.TransformerException: org.xml.sax.SAXParseException: 在元素类型"xsl:template"后面必须指定属性、">"或"/>"表明未能读取 LDAPTransformation.xml 文件。请检查人员分配配置和所配置的 XSLT 文件是否有错误。
- trace.log 文件中的找不到 LDAP 对象。DN: uid=unknown,cn=users,dc=ibm,dc=com [LDAP: 错误代码 32 - 没有这

样的对象] 表明找不到 LDAP 条目。请在任务模型中检查任务模型的人员分配条件（动词）参数是否与 LDAP 目录内容不匹配。

- trace.log 文件中的在 `uid=test222,cn=users,dc=ibm,dc=com` 中找不到所请求的属性"uid"表明在所查询的 LDAP 对象中找不到某个属性。请在任务模型中检查任务模型的人员分配条件（动词）参数是否与 LDAP 目录内容不匹配。并检查人员分配配置的 XSLT 文件是否有错误。

与组工作项和“Group”人员分配条件相关的问题

在使用 Group 人员分配条件时，可能会发生下列情况：

- 虽然指定了组名，但未对组成员进行授权：
 - 使用“本地操作系统”注册表启用 WebSphere 安全性时，请指定组的短名称；使用 LDAP 注册表时，请指定组 dn。
 - 确保遵循组名区分大小写的规则。

发生此情况的一种可能原因是，已经对 WebSphere 安全性配置了 LDAP 用户注册表，并且选中授权时忽略大小写选项。在这种情况下，请取消选择此选项，或者指定全大写的 LDAP 组 dn。

- 对组成员资格所作的更改未立即反映在权限中。如果受影响的用户仍处于登录状态，那么将发生这种情况。用户的组成员资格高速缓存在登录会话中，而缺省情况下此会话将在两小时后到期。您可以等待该登录会话到期（缺省到期时间为两小时），也可以重新启动应用程序服务器。人员任务管理器提供的刷新方法不适用于此人员分配条件。请注意，流程启动者的组成员资格列表永远不会被刷新。

清除已存储的人员分配结果

人员分配结果存储在数据库中。执行人员分配刷新时，将刷新所有已存储的人员分配结果。在删除致使计算人员分配结果的任务实例所在的任务模板时，还将删除已存储的人员分配结果。但是，如果仅删除正在使用已存储的人员分配结果的任务实例，那么不会删除已存储的人员分配结果。

为了避免在数据库中存储大量不必要的人员分配结果，请在任务模板的上下文中执行下列步骤：

1. 评估人员分配条件定义是将生成共享的还是非共享的人员分配结果。
2. 对于非共享的分配结果，请考虑设置用于清除人员分配结果的清除过程。清除时间间隔应该基于预期的任务实例数以及每个清除时间间隔内的非共享人员分配结果数。有关如何应用基于脚本的清除过程的更多信息，请参阅使用管理命令除去未使用的人员查询结果。

相关参考

第 567 页的『启用替代功能后，任务容器无法启动』

您已创建新的业务流程编排器配置并启用替代功能，或者已对现有配置启用替代功能。

业务流程编排器资源管理器故障诊断

使用本主题来解决与业务流程编排器资源管理器相关的问题。

关于此任务

使用以下信息来解决与业务流程编排器资源管理器相关的问题。

- 如果您尝试使用浏览器来访问业务流程编排器资源管理器，但接收到错误消息而不是登录页面，那么请尝试执行下列操作：
 - 使用管理控制台，确保服务器上确实已部署了 Web 客户端应用程序 `BPCExplorer_node_name_server_name` 并且该应用程序处于运行状态。
 - 在管理控制台中的应用程序页面上，在“查看部署描述符”中验证上下文根是否是您设置业务流程编排器资源管理器时使用的上下文根。
- 如果使用业务流程编排器资源管理器时接收到错误消息，那么在错误页面上单击**搜索更多信息**链接。这将开始在 IBM 技术支持站点上搜索错误代码。此站点仅提供英文版资料。请将业务流程编排器资源管理器错误页面中显示的错误消息代码复制到剪贴板。错误代码的格式为 `CWWBcnnnc`，其中每个 `c` 是一个字符，`nnnn` 是一个 4 位数。转到 WebSphere Process Server 技术支持页面。将该错误代码粘贴到 **Additional search terms** 字段中，然后单击 **Go**。
- 如果发生 `StandardFaultException` 异常并且标准故障为 `missingReply`（消息 `CWWBE0071E`），那么此为流程模型问题的症状。有关如何解决此问题的更多信息，请参阅第 578 页的『对业务流程和人员任务的管理进行故障诊断』。
- 如果您可以登录到业务流程编排器资源管理器，但某些项未显示，或者某些操作未处于启用状态，那么表明权限有问题。

此问题的可能解决方案包括：

- 使用管理控制台来确保已启用 WebSphere 管理安全性。
- 检查您是否使用了正确的标识登录到业务流程编排器资源管理器。如果使用非流程管理员或任务管理员的用户标识登录，那么所有管理视图和选项都将不可视或处于未启用状态。
- 使用 WebSphere Integration Developer 检查或修改业务流程中定义的权限设置。
- 错误消息 `CWWBU0001E`：“调用 `BFMConnection` 函数时发生了通信错误”或“调用 `HTMConnection` 函数时发生了通信错误”。此错误表明业务流程容器或人员任务容器已停止，并且客户机未能连接到服务器。请验证业务流程容器和人员任务容器是否处于运行状态并且可访问。嵌套异常可能包含有关此问题的更多详细信息。
- 错误消息 `WWBU0024E`：未能与本地业务流程 EJB 建立连接，原因为“命名异常”。业务流程容器未运行时，如果用户尝试登录，那么将抛出此错误。请验证应用程序 `BPEContainer_InstallScope` 是否处于运行状态，其中 `InstallScope` 是 `cluster_name` 或 `hostname_servername`。

相关任务

第 572 页的『对业务流程的执行进行故障诊断』
本节描述常见业务流程执行问题的解决方案。

业务流程编排器观察器故障诊断

如果您使用业务流程编排器观察器时遇到问题，请参阅本主题中的信息。

症状

业务流程编排器观察器的欢迎页面未显示任何事件。

原因与解决方案

业务流程编排器观察器的数据库未包含任何事件，或者尚未变换任何事件。

原因	解决方案
未启用 CEI 日志记录。	请确保对业务流程容器启用 CEI 日志记录。请参阅对业务流程编排器观察器启用日志记录以启用 CEI 日志记录。
公共事件基础结构事件服务器或业务流程编排器事件收集器未处于运行状态。	使用管理控制台，检查公共事件基础结构事件服务器和业务流程编排器事件收集器是否处于运行状态。
已对业务流程禁用事件监视功能。	确保在 WebSphere Integration Developer 中的流程模型定义中启用事件监视功能。有关如何对业务流程启用事件监视功能的建议，请参阅 WebSphere Integration Developer 信息中心。
未触发事件变换器。	请重新启动业务流程编排器事件收集器以触发事件变换器。
业务流程编排器事件收集器的配置设置不合适，导致数据在观察器中不可视。	请调用 <code>setupEventCollector</code> 配置脚本，以更改业务流程编排器事件收集器配置设置 <code>BPCEventTransformerEventCount</code> 、 <code>BPCEventTransformerMaxWaitTime</code> 和 <code>BPCEventTransformerToleranceTime</code> 。要了解如何更改业务流程编排器事件收集器的配置设置，请参阅更改配置参数。

如果问题仍存在

- 在服务器的系统日志文件 `SystemOut.log` 中检查错误消息。
- 检查业务流程编排器事件收集器和业务流程编排器观察器的部署及配置。要检查配置设置，请调用 `setupEventCollector` 和 `setupObserver` 配置脚本。有关如何更改业务流程编排器事件收集器配置设置的更多信息，请参阅第 234 页的『更改业务流程编排器观察器的配置参数』。
- 在管理控制台中，对观察器组件启用跟踪工具：**日志记录与跟踪** → **server1** → **诊断跟踪服务** → **更改日志详细信息级别**。将 `com.ibm.bpe.observer.*` 的详细信息级别设置为 `all`，然后重新启动 `BPCECollector` 和 `BPCObserver` 应用程序。

使用与流程相关的审计跟踪信息以及与任务相关的审计跟踪信息

本主题阐述业务流程和人员任务的事件类型及数据库结构。

开始之前

必须对业务流程容器和/或任务容器启用日志记录。

关于此任务

如果已启用日志记录，那么对于业务流程或人员任务的每个导航步骤，将把信息写入审计日志或公共事件基础结构（CEI）日志。有关 CEI 的更多信息，请参阅《监视 WebSphere Process Server》PDF。下列主题描述了业务流程和人员任务的事件类型及数据库结构。

业务流程的审计事件类型

本节描述业务流程处理期间可能会写入审计日志的事件的类型。

要记录事件，必须符合下列条件：

- 对业务流程容器启用了相应的审计日志记录类型
- 必须对流程模型中的相应实体启用此事件

下列各表列示了业务流程运行期间可能发生的审计事件的代码。

表 94. 流程实例事件

审计事件	事件码
PROCESS_STARTED	21000
PROCESS_SUSPENDED	21001
PROCESS_RESUMED	21002
PROCESS_COMPLETED	21004
PROCESS_TERMINATED	21005
PROCESS_RESTARTED	21019
PROCESS_DELETED	21020
PROCESS_FAILED	42001
PROCESS_COMPENSATING	42003
PROCESS_COMPENSATED	42004
PROCESS_TERMINATING	42009
PROCESS_FAILING	42010
PROCESS_CORRELATION_SET_INITIALIZED	42027
PROCESS_COMPENSATION_INDOUBT	42030
PROCESS_WORKITEM_DELETED	42041
PROCESS_WORKITEM_CREATED	42042
PROCESS_COMPENSATION_FAILED	42046
PROCESS_EVENT_RECEIVED	42047
PROCESS_EVENT_ESCALATED	42049
PROCESS_WORKITEM_TRANSFERRED	42056
PROCESS_PARTNER_CHANGED	42058

表 95. 活动事件

审计事件	事件码
ACTIVITY_READY	21006
ACTIVITY_STARTED	21007
ACTIVITY_COMPLETED	21011
ACTIVITY_CLAIM_CANCELED	21021
ACTIVITY_CLAIMED	21022
ACTIVITY_TERMINATED	21027
ACTIVITY_FAILED	21080
ACTIVITY_EXPIRED	21081

表 95. 活动事件 (续)

审计事件	事件码
ACTIVITY_LOOPED	42002
ACTIVITY_SKIPPED	42005
ACTIVITY_TERMINATING	42008
ACTIVITY_FAILING	42011
ACTIVITY_OUTPUT_MESSAGE_SET	42012
ACTIVITY_FAULT_MESSAGE_SET	42013
ACTIVITY_STOPPED	42015
ACTIVITY_FORCE_RETRIED	42031
ACTIVITY_FORCE_COMPLETED	42032
ACTIVITY_UNDO_STARTED	42033
ACTIVITY_UNDO_SKIPPED	42034
ACTIVITY_UNDO_COMPLETED	42035
ACTIVITY_MESSAGE_RECEIVED	42036
ACTIVITY_LOOP_CONDITION_TRUE	42037
ACTIVITY_LOOP_CONDITION_FALSE	42038
ACTIVITY_WORKITEM_DELETED	42039
ACTIVITY_WORKITEM_CREATED	42040
ACTIVITY_ESCALATED	42050
ACTIVITY_WORKITEM_REFRESHED	42054
ACTIVITY_WORKITEM_TRANSFERRED	42055
ACTIVITY_PARALLEL_BRANCHES_STARTED	42057

表 96. 与变量相关的事件

审计事件	事件码
VARIABLE_UPDATED	21090

表 97. 控制链路事件

审计事件	事件码
LINK_EVALUATED_TO_TRUE	21034
LINK_EVALUATED_TO_FALSE	42000

表 98. 流程模板事件

审计事件	事件码
PROCESS_INSTALLED	42006
PROCESS_UNINSTALLED	42007

表 99. 作用域实例事件

审计事件	事件码
SCOPE_STARTED	42020
SCOPE_SKIPPED	42021

表 99. 作用域实例事件 (续)

审计事件	事件码
SCOPE_FAILED	42022
SCOPE_FAILING	42023
SCOPE_TERMINATED	42024
SCOPE_COMPLETED	42026
SCOPE_COMPENSATING	42043
SCOPE_COMPENSATED	42044
SCOPE_COMPENSATION_FAILED	42045
SCOPE_EVENT_RECEIVED	42048
SCOPE_EVENT_ESCALATED	42051

人员任务的审计事件类型

本节描述人员任务处理期间可能会写入审计日志的事件的类型。

要记录事件，必须符合下列条件：

- 对人员任务容器启用了相应的审计日志记录类型
- 必须对任务模型中的相应实体启用此事件

下列各表列示了人员任务运行期间可能发生的审计事件的代码。

表 100. 任务实例事件

审计事件	事件码
TASK_CREATED	51001
TASK_DELETED	51002
TASK_STARTED	51003
TASK_COMPLETED	51004
TASK_CLAIM_CANCELLED	51005
TASK_CLAIMED	51006
TASK_TERMINATED	51007
TASK_FAILED	51008
TASK_EXPIRED	51009
TASK_WAITING_FOR_SUBTASK	51010
TASK_SUBTASKS_COMPLETED	51011
TASK_RESTARTED	51012
TASK_SUSPENDED	51013
TASK_RESUMED	51014
TASK_COMPLETED_WITH_FOLLOW_ON	51015
TASK_UPDATED	51101
TASK_OUTPUT_MESSAGE_UPDATED	51103
TASK_FAULT_MESSAGE_UPDATED	51104
TASK_WORKITEM_DELETED	51201
TASK_WORKITEM_CREATED	51202

表 100. 任务实例事件 (续)

审计事件	事件码
TASK_WORKITEM_TRANSFERRED	51204
TASK_WORKITEM_REFRESHED	51205

表 101. 任务模板事件

审计事件	事件码
TASK_TEMPLATE_INSTALLED	52001
TASK_TEMPLATE_UNINSTALLED	52002

表 102. 升级实例事件

审计事件	事件码
ESCALATION_FIRED	53001
ESCALATION_WORKITEM_DELETED	53201
ESCALATION_WORKITEM_CREATED	53202
ESCALATION_WORKITEM_TRANSFERRED	53204
ESCALATION_WORKITEM_REFRESHED	53205

业务流程的审计跟踪数据库视图结构

AUDIT_LOG_B 数据库视图提供有关业务流程的审计日志信息。

要阅读审计跟踪的内容，请使用 SQL 或任何其他支持读取数据库表和视图的管理工具。

审计事件与流程实体相关。审计事件类型依赖于该事件所引用的实体。审计事件类型包括：

- 流程模板事件 (PTE)
- 流程实例事件 (PIE)
- 活动实例事件 (AIE)
- 与变量相关的事件 (VAR)
- 控制链路事件 (CLE)
- 与作用域相关的事件 (SIE)

要获取审计事件类型码列表，请参阅第 587 页的『业务流程的审计事件类型』。

下表描述了 AUDIT_LOG_B 审计跟踪视图的结构。此表列示了列名、事件类型以及该列的简要描述。

直接插入任务记录在 AUDIT_LOG_B 审计跟踪视图中，而不是记录在 TASK_LOG 审计跟踪视图中。例如，声明直接插入参与任务时将生成 ACTIVITY_CLAIMED 事件，而不会生成与该任务相关的事件。

表 103. AUDIT_LOG_B 审计跟踪视图的结构

名称	PTE	PIE	AIE	VAR	CLE	SIE	描述
AIID			x				与当前事件相关的活动实例的标识。
ALID	x	x	x	x	x	x	审计日志条目的标识。

表 103. AUDIT_LOG_B 审计跟踪视图的结构 (续)

名称	PTE	PIE	AIE	VAR	CLE	SIE	描述
EVENT_TIME	x	x	x	x	x	x	事件发生时间的的时间戳记, 采用全球标准时间 (UTC) 格式。
EVENT_TIME_UTC	x	x	x	x	x	x	事件发生时间的的时间戳记, 采用全球标准时间 (UTC) 格式。
AUDIT_EVENT	x	x	x	x	x	x	所发生的事件的类型。
PTID	x	x	x	x	x	x	与当前事件相关的流程的流程模板标识。
PIID		x	x	x	x	x	与当前事件相关的流程实例的流程实例标识。
VARIABLE_NAME				x			与当前事件相关的变量的名称。
SIID						x	与事件相关的作用域实例的标识。
PROCESS_TEMPL_NAME	x	x	x	x	x	x	与当前事件相关的流程模板的流程模板名称。
TOP_LEVEL_PIID		x	x	x	x	x	与当前事件相关的顶层流程的标识。
PARENT_PIID		x	x	x	x	x	父流程的流程实例标识, 如果不存在父流程, 那么为 null。
VALID_FROM	x	x	x	x	x	x	与当前事件相关的流程模板的生效日期。
VALID_FROM_UTC	x	x	x	x	x	x	与当前事件相关的流程模板的生效日期, 采用全球标准时间 (UTC) 格式。
ATID			x				与当前事件相关的活动模板的标识。
ACTIVITY_NAME			x			x	所发生的事件的相关活动名称。

表 103. AUDIT_LOG_B 审计跟踪视图的结构 (续)

名称	PTE	PIE	AIE	VAR	CLE	SIE	描述
ACTIVITY_KIND			x				<p>所发生的事件的相关活动类型。可能的值是:</p> <p>KIND_EMPTY 3 KIND_INVOKE 21 KIND_RECEIVE 23 KIND_REPLY 24 KIND_THROW 25 KIND_TERMINATE 26 KIND_WAIT 27 KIND_COMPENSATE 29 KIND_SEQUENCE 30 KIND_SWITCH 32 KIND_WHILE 34 KIND_PICK 36 KIND_FLOW 38 KIND_SCRIPT 42 KIND_STAFF 43 KIND_ASSIGN 44 KIND_CUSTOM 45 KIND_RETHROW 46 KIND_FOR_EACH_SERIAL 47 KIND_FOR_EACH_PARALLEL 49</p> <p>这些值是为 ActivityInstanceData.KIND_* 定义的常量。</p>
ACTIVITY_STATE			x				<p>与事件相关的活动的状态。可能的值是:</p> <p>STATE_INACTIVE 1 STATE_READY 2 STATE_RUNNING 3 STATE_SKIPPED 4 STATE_FINISHED 5 STATE_FAILED 6 STATE_TERMINATED 7 STATE_CLAIMED 8 STATE_TERMINATING 9 STATE_FAILING 10 STATE_WAITING 11 STATE_EXPIRED 12 STATE_STOPPED 13</p> <p>这些值是为 ActivityInstanceData.STATE_* 定义的常量。</p>
CONTROL_LINK_NAME					x		与当前链接事件相关的链接的名称。

表 103. AUDIT_LOG_B 审计跟踪视图的结构 (续)

名称	PTE	PIE	AIE	VAR	CLE	SIE	描述
PRINCIPAL		x	x	x	x	x	主体的名称。对于 PROCESS_DELETED 事件而言，不会设置此列。
VARIABLE_DATA				x			已更新变量事件的变量数据。
EXCEPTION_TEXT		x	x			x	导致活动或流程失败的异常消息。适用于： PROCESS_FAILED ACTIVITY_FAILED SCOPE_FAILED
DESCRIPTION		x	x	x	x	x	活动或流程的描述，包含可能已解析的替换变量。
CORR_SET_INFO		x					在流程启动时初始化的关联集的字符串表示。与 processCorrelationSetInitialized 事件（42027）一起提供。
USER_NAME		x	x				已更改了工作项的用户的名称。此属性适用于下列事件： <ul style="list-style-type: none"> • 流程实例工作项已删除 • 活动实例工作项已删除 • 流程实例工作项已创建 • 活动实例工作项已创建
ADDITIONAL_INFO		x	x			x	此字段的内容取决于事件类型： ACTIVITY_WORKITEM_TRANSFERRERD 和 PROCESS_WORK_ITEM_TRANSFERRERD 接收工作项的用户的名称。 ACTIVITY_WORKITEM_CREATED 、 ACTIVITY_WORKITEM_REFRESHED 和 ACTIVITY_ESCALATED “;”分隔的用户列表，已经为此列表中的所有用户创建或刷新了工作项。如果此列表仅包含一个用户，那么此用户的用户名将填入 USER_NAME 字段，并且 ADDITIONAL_INFO 字段为空，即包含 NULL。 PROCESS_EVENT_RECEIVED 和 SCOPE_EVENT_RECEIVED 如果能够获得事件处理程序所接收的操作的类型，那么此字段将包含该类型。使用以下格式：“{”端口类型名称空间“}”端口类型名“:”操作名。对于“onAlarm”事件而言，不会设置此字段。

人员任务的审计跟踪数据库视图结构

TASK_AUDIT_LOG 数据库视图提供有关人员任务的审计日志信息。

直接插入任务记录在 AUDIT_LOG_B 视图中。所有其他类型的任务记录在 TASK_AUDIT_LOG 视图中。

要阅读审计跟踪的内容，请使用 SQL 或任何其他支持读取数据库表和视图的管理工具。

审计事件与任务实体相关。审计事件类型依赖于该事件所引用的实体。审计事件类型包括：

- 任务实例事件 (TIE)
- 任务模板事件 (TTE)
- 升级实例事件 (EIE)

下表描述了 TASK_AUDIT_LOG 审计跟踪视图的结构。此表列示了列名、事件类型以及该列的简要描述。

直接插入任务记录在 AUDIT_LOG_B 审计跟踪视图中，而不是记录在 TASK_AUDIT_LOG 审计跟踪视图中。例如，声明直接插入参与任务时将生成 ACTIVITY_CLAIMED 事件，而不会生成与该任务相关的事件。

表 104. TASK_AUDIT_LOG 审计跟踪视图的结构

名称	TIE	TTE	EIE	描述
ALID	x	x	x	审计日志条目的标识。
AUDIT_EVENT	x	x	x	所发生的事件的类型。要获取审计事件码列表，请参阅第 589 页的『人员任务的审计事件类型』。
CONTAINMENT_CTX_ID	x	x		外围上下文的标识，例如 ACOID、PTID 或 PIID。
DESCRIPTION	x		x	已解析的描述字符串，其中，描述字符串中的占位符已替换为当前值。所有受影响的语言以 XML 文档格式一起记录在此列中。将只记录描述字符串包含创建等事件的占位符或者已经根据更新等事件进行明确更新的语言。
ESIID			x	与当前事件相关的升级实例的标识。
ESTID			x	与当前事件相关的升级模板的标识。
EVENT_TIME	x	x	x	事件的发生时间，采用全球标准时间 (UTC) 格式。
FAULT_NAME	x			故障消息的名称。此属性适用于下列事件： TASK_FAILED TASK_FAULT_MESSAGE_UPDATED
FAULT_NAME_SPACE	x			故障消息类型的名称空间。此属性适用于下列事件： TASK_FAILED TASK_FAULT_MESSAGE_UPDATED
FOLLOW_ON_TKIID	x			后续任务实例的标识。
MESSAGE_DATA	x			新创建或更新的输入消息、输出消息或故障消息的内容。
NAME	x	x	x	与事件相关的任务实例、任务模板或升级实例的名称。

表 104. TASK_AUDIT_LOG 审计跟踪视图的结构 (续)

名称	TIE	TTE	EIE	描述
NAMESPACE	x	x		与事件相关的任务实例、任务模板或升级实例的名称空间。
NEW_USER				所移交或创建的工作项的新所有者。如果此值是通过 USERS 字段提供的, 那么可能为 null。另请参阅 USERS 字段。此属性适用于下列事件:
	x			TASK_WORKITEM_CREATED
	x			TASK_WORKITEM_TRANSFERRED
			x	ESCALATION_WORKITEM_CREATED
			x	ESCALATION_WORKITEM_TRANSFERRED
OLD_USER				所移交的工作项的先前所有者。此属性适用于下列事件:
	x			TASK_WORKITEM_TRANSFERRED
	x			TASK_WORKITEM_DELETED
			x	ESCALATION_WORKITEM_TRANSFERRED
			x	ESCALATION_WORKITEM_DELETED
PARENT_CONTEXT_ID	x			任务的父上下文(例如活动模板或任务实例)的标识。仅对于子任务和后续任务, 才会设置此属性。
PARENT_TASK_NAME	x			父任务实例或父任务模板的名称。仅对于子任务和后续任务, 才会设置此属性。
PARENT_TASK_NAMESP	x			父任务实例或父任务模板的名称空间。仅对于子任务和后续任务, 才会设置此属性。
PARENT_TKIID	x			父任务实例的标识。
PRINCIPAL	x	x	x	主体的名称, 此主体的请求触发了事件。
TASK_KIND	x	x		任务的种类。可能的值是: KIND_HUMAN 101 KIND_ORIGINATING 103 KIND_PARTICIPATING 105 KIND_ADMINISTRATIVE 106
TASK_STATE	x			任务或任务模板的状态。任务模板的可能值是: STATE_STARTED 1 STATE_STOPPED 2 任务实例的可能值是: '1': 'STATE_INACTIVE' '2': 'STATE_READY' '3': 'STATE_RUNNING' '5': 'STATE_FINISHED' '6': 'STATE_FAILED' '7': 'STATE_TERMINATED' '8': 'STATE_CLAIMED' '12': 'STATE_EXPIRED' '101': 'FORWARDED'
TKIID	x		x	任务实例的标识。

表 104. TASK_AUDIT_LOG 审计跟踪视图的结构 (续)

名称	TIE	TTE	EIE	描述
TKTID	x	x		任务模板的标识。
TOP_TKIID	x			顶层任务实例的标识。
USERS	x		x	对任务或升级工作项指定的新用户标识。如果此值是通过 NEW_USER 字段提供的, 那么可能为 null。请参阅 NEW_USER 字段以获取此属性所适用于的事件列表。
VALID_FROM		x		与当前事件相关的任务模板的生效日期。
WORK_ITEM_REASON	x		x	<p>此工作项的分配原因。可能的值是:</p> <p>POTENTIAL_OWNER 1 EDITOR 2 READER 3 OWNER 4 POTENTIAL_STARTER 5 STARTER 6 ADMINISTRATOR 7 POTENTIAL_SENDER 8 ORIGINATOR 9 ESCALATION_RECEIVER 10 POTENTIAL_INSTANCE_CREATOR 11</p> <p>将为所有与工作项相关的事件设置原因: 对于与升级工作项相关的事件, 将设置 ESCALATION_RECEIVER, 其他原因适用于与任务工作项相关的事件。</p>

第 8 部分 附属资料

声明

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.*

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

*IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan*

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区： International Business Machines Corporation“按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。

凡这些样本程序的每份拷贝或其任何部分或任何演绎作品，都必须包括如下版权声明：©（贵公司的名称）（年）。此部分代码是根据 IBM Corp. 的样本程序衍生出来的。
© Copyright IBM Corp.（输入年份）。All rights reserved.

如果您正在查看本信息的软拷贝，图片和彩色图例可能无法显示。

编程接口信息

如果提供了编程接口信息，那么该信息旨在帮助您创建使用此程序的应用程序软件。

通用编程接口允许您编写的应用程序软件包含由此程序的工具提供的服务。

然而，本信息还可能包含诊断、修改和调整信息。这些诊断、修改和调整信息用于帮助您调试应用软件。

警告： 不要将此诊断、修改和调整信息用作编程接口，因为它是会更改的。

商标和服务标记

IBM、IBM 徽标、AIX、DB2、i5/OS、Informix、iSeries、MQSeries、WebSphere 和 z/OS 是 International Business Machines Corporation 在美国和/或其他国家或地区的注册商标, Cloudscape、DB2 Connect 和 DB2 Universal Database 是该公司在美国和/或其他国家或地区的商标。

Java 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和/或其他国家或地区的商标。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和/或其他国家或地区的注册商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的注册商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。

此产品包含由 Eclipse Project (<http://www.eclipse.org>) 开发的软件。



IBM WebSphere Process Server for Multiplatforms V6.1.0



中国印刷