

Workday Studio

Product Summary

March 9, 2026



内容

设置时需考虑的事项：Workday Studio.....	10
Studio 基本功能.....	12
下载并安装 Windows 版 Workday Studio.....	12
下载并安装 macOS 版 Workday Studio.....	14
概念：Workday Studio.....	15
概念：集成开发生命周期.....	17
概念：MediationContext 对象.....	17
概念：集成系统和事件.....	18
参考信息：界面基础知识.....	18
集成设计.....	19
程序集.....	19
创建程序集项目.....	19
向构成添加步骤.....	20
添加子元素.....	21
连接程序集元素.....	21
配置程序集元素属性.....	22
管理集合和项目.....	23
声明命名空间前缀.....	23
创建路径变量.....	24
概念：程序集 Palette.....	24
概念：构成和步骤.....	26
概念：程序集注释.....	26
概念：程序集项目的链接资源文件.....	27
概念：“Quick Outline”视图.....	27
参考信息：程序集属性.....	27
参考信息：泳道属性.....	28
参考信息：连接线属性.....	29
程序集传输.....	29
在 Workday Studio 中将枚举用作启动参数.....	29
概念：程序集传输.....	29
概念：枚举.....	31
参考信息：Workday-In 传输属性.....	32
参考信息：Local-In 传输属性.....	33
参考信息：Workday-Out-Rest 传输属性.....	34
参考信息：Workday-Out-Soap 传输属性.....	35
参考信息：Local-Out 传输属性.....	36
参考信息：HTTP-Out 传输属性.....	37
参考信息：FTP-Out 传输属性.....	40
参考信息：FTPS-Out 传输属性.....	43
参考信息：SFTP-Out 传输属性.....	47
参考信息：XMPP-Out 传输属性.....	50
参考信息：Custom-Out 传输属性.....	51
参考信息：Email-Out 传输属性.....	53
HTTP-Out 传输子元素.....	59
概念：HTTP-Out 传输子元素.....	59

参考信息：HTTP-Basic-Auth 属性.....	59
参考信息：HTTP-Custom-Auth 属性.....	60
参考信息：HTTPS 属性.....	61
FTPS-Out 传输子元素.....	61
概念：FTPS-Out 传输子元素.....	61
参考信息：Proxy-Properties 属性.....	62
参考信息：Firewall-Properties 属性.....	62
参考信息：Client-Key 属性.....	62
参考信息：Server-Key 属性.....	63
SFTP-Out 子元素.....	63
概念：SFTP-Out 子元素.....	63
参考信息：Proxy-Properties 属性.....	63
参考信息：Private-Key 属性.....	64
构成.....	64
概念：构成.....	64
参考信息：Async-Mediation 构成属性.....	65
参考信息：Sync-Mediation 构成属性.....	65
参考信息：Custom-Mediation 构成属性.....	66
参考信息：Route 构成属性.....	67
参考信息：Splitter 构成属性.....	67
参考信息：Aggregator 构成属性.....	67
Route 构成子元素.....	68
概念：Route 构成子元素.....	68
参考信息：Custom-Strategy 属性.....	68
参考信息：Regex-Strategy 属性.....	69
参考信息：MVEL-Strategy 属性.....	69
参考信息：Round-Robin-Strategy 属性.....	69
参考信息：XPath-Strategy 属性.....	69
参考信息：Failover-Strategy 属性.....	70
参考信息：All-Strategy 属性.....	70
参考信息：Doc-Iterator 属性.....	71
参考信息：循环策略属性.....	71
参考信息：Choose-Route 属性.....	71
参考信息：Sub-Route 属性.....	72
Splitter 构成子元素.....	72
概念：Splitter 构成子元素.....	72
参考信息：Custom-Splitter 属性.....	73
参考信息：Standard-Splitter 属性.....	73
参考信息：XPath-Splitter 属性.....	73
参考信息：XML-Stream-Splitter 属性.....	73
参考信息：JSON-Splitter 属性.....	74
参考信息：Unzip-Splitter 属性.....	74
参考信息：Mtable-Splitter 属性.....	74
Standard-Splitter 子元素.....	74
概念：Standard-Splitter 子元素.....	74
参考信息：Header-Ends-With 属性.....	75
参考信息：Header-Fixed-Lines 属性.....	75
参考信息：Content-Fixed-Lines 属性.....	75
参考信息：Content-Starts-Ends 属性.....	75
参考信息：Content-Starts-With 属性.....	76
参考信息：Footer-Fixed-Lines 属性.....	76
参考信息：Footer-Starts-With 属性.....	76
Aggregator 构成子元素.....	76
概念：Aggregator 构成子元素.....	76
参考信息：Custom-Batch-Strategy 属性.....	77
参考信息：Size-Batch-Strategy 属性.....	77

参考信息：Time-Batch-Strategy 属性.....	77
参考信息：Custom-Collater 属性.....	77
参考信息：Message-Content-Collater 属性.....	78
参考信息：XML-Message-Content-Collater 属性.....	78
参考信息：JSON-Aggregator 属性.....	79
参考信息：Zip-File-Collater 属性.....	80
参考信息：Mtable-Collater 属性.....	80
通用构成.....	81
概念：通用构成.....	81
参考信息：FTP 子程序集属性.....	83
参考信息：GetEventConfigurations 子程序集属性.....	85
参考信息：GetEventDocuments 子程序集属性.....	87
参考信息：GetIntegrationEvent 子程序集属性.....	88
参考信息：GetIntegrationSystems 子程序集属性.....	90
参考信息：PagedGet 子程序集属性.....	91
参考信息：PagedGetLocalPaging 子程序集属性.....	94
参考信息：PdfPrintStep 子程序集属性.....	96
参考信息：PutIntegrationEvent 子程序集属性.....	98
参考信息：PutIntegrationMessage 子程序集属性.....	99
参考信息：SalesforceConnector 子程序集属性.....	102
PrismAnalytics 子组件.....	103
错误处理程序构成.....	114
概念：错误处理程序构成.....	114
参考信息：Custom-Error-Handler 构成属性.....	115
参考信息：Log-Error 构成属性.....	116
参考信息：Send-Error 构成属性.....	117
转换步骤.....	118
使用文本架构文件将文本转换为 XML.....	118
概念：转换步骤.....	121
参考信息：XSLT 步骤属性.....	122
参考信息：XSLT+ 步骤属性.....	124
参考信息：FOP 步骤属性.....	126
参考信息：Text-Excel 步骤属性.....	129
参考信息：TextSchema 步骤属性.....	131
参考信息：Wrap-Soap 步骤属性.....	132
参考信息：XML-to-Java 步骤属性.....	133
参考信息：JSON 转换器属性.....	135
参考信息：XML-to-JSON 步骤属性.....	136
参考信息：JSON-to-XML 步骤属性.....	138
参考信息：Java-to-XML 步骤属性.....	139
参考信息：XML-to-CSV 步骤属性.....	140
参考信息：CSV-to-XML 步骤属性.....	142
参考信息：Character-Conversion 步骤属性.....	144
参考信息：Mtable-Builder 步骤属性.....	146
参考信息：Mtable-Writer 步骤属性.....	147
参考信息：ETV 步骤属性.....	148
参考信息：XTT 步骤属性.....	149
Mtable-Builder 子元素.....	150
概念：Mtable-Builder 子元素.....	150
参考信息：CSV-Mtable-Reader 属性.....	150
参考信息：JSON-Mtable-Reader 属性.....	151
Mtable-Writer 子元素.....	151
概念：Mtable-Writer 子元素.....	151
参考信息：CSV-Mtable-Writer 属性.....	152
参考信息：JSON-Mtable-Writer 属性.....	152
日志记录和事件记录步骤.....	152

概念：日志记录和事件记录步骤.....	152
参考信息：Log 步骤属性.....	153
参考信息：Cloud-Log 步骤属性.....	154
消息操作步骤.....	156
概念：消息操作步骤.....	156
参考信息：Copy 步骤属性.....	156
参考信息：JavaScript 步骤属性.....	158
参考信息：Set-Headers 步骤属性.....	160
参考信息：Write 步骤属性.....	161
验证步骤.....	162
概念：验证步骤.....	162
参考信息：Eval 步骤属性.....	163
参考信息：Validate 步骤属性.....	163
参考信息：Validate-Exp 步骤属性.....	165
参考信息：Validate-Xpath 步骤属性.....	166
编码步骤.....	168
概念：编码步骤.....	168
概念：PGP 文件名.....	168
参考信息：PGP-Encrypt 步骤属性.....	169
参考信息：PGP-Decrypt 步骤属性.....	172
参考信息：Base64-Decode 步骤属性.....	173
参考信息：Base64-Encode 步骤属性.....	175
参考信息：Zip 步骤属性.....	176
参考信息：Unzip 步骤属性.....	177
参考信息：Compress 步骤属性.....	179
参考信息：Decompress 步骤属性.....	180
其他步骤.....	182
概念：其他步骤.....	182
参考信息：Custom 步骤属性.....	182
参考信息：Set-Dynamic-Endpoint 步骤属性.....	184
参考信息：Store 步骤属性.....	184
参考信息：Retrieve 步骤属性.....	187
参考信息：Xmldiff 步骤属性.....	188
参考信息：Enqueue-Message 步骤属性.....	191
程序集模块.....	193
创建程序集模块.....	193
将程序集模块添加到项目.....	194
概念：程序集模块.....	195
子程序集.....	195
创建子程序集.....	195
概念：子程序集.....	196
自定义程序集构成.....	196
创建自定义 Java 类和 Spring Bean.....	196
概念：自定义程序集元素.....	197
概念：自定义元素封装.....	199
Spring Bean 配置.....	200
概念：Spring Bean 配置.....	200
参考信息：全局 Bean 属性.....	201
参考信息：单个 Bean 属性.....	201
参考信息：别名属性.....	202
参考信息：Property 属性.....	202
参考信息：Constructor-Arg 属性.....	203
参考信息：Ref 属性.....	203
参考信息：列表属性.....	203
参考信息：映射属性.....	203
Message Builder.....	204

基于架构或 WSDL 创建 XML 字面量.....	204
基于架构或 WSDL 创建 XPath 表达式.....	205
概念：Workday Studio 消息构建器.....	205
参考信息：消息元素属性.....	206
传输绑定.....	209
概念：传输绑定.....	209
参考信息：WSDL-HTTP-Binding 属性.....	210
参考信息：WSDL-SOAP-Binding 属性.....	210
WSDL-SOAP-Binding 子元素.....	211
概念：WSDL-SOAP-Binding 子元素.....	211
参考信息：Custom-Policy 属性.....	211
参考信息：WS-Rm 属性.....	211
参考信息：WS-Security 属性.....	212
参考信息：WS-Ut 属性.....	212
Schema Explorer.....	212
显示 WSDL 和 XSD 文件.....	212
显示 Workday Web 服务 WSDL.....	213
显示自定义 RaaS 报告架构.....	213
概念：Schema Explorer.....	214
MVEL.....	214
概念：MVEL 基础知识.....	214
概念：MVEL 与 Workday Studio.....	217
参考信息：Workday Studio 帮助程序对象.....	217
参考信息：文档访问器帮助程序对象.....	221
参考信息：启动参数帮助程序对象.....	224
参考信息：Util 帮助程序对象.....	226
参考信息：Workday Studio MVEL 表达式示例.....	228
参考信息：Workday Studio MVEL 自动转换语法示例.....	230
参考信息：MVEL 方法.....	232
Workday Studio 集成服务.....	233
创建属性映射服务.....	233
创建传送服务.....	235
创建侦听器服务.....	235
创建检索服务.....	236
创建序列生成器服务.....	236
创建事务日志服务.....	237
创建报告服务.....	238
创建自定义对象服务.....	239
创建数据初始化服务.....	240
创建服务引用.....	240
在 Workday-Out-Rest 传输上添加对自定义报告的引用.....	241
在 Workday-Out-Rest 传输上添加对自定义对象的引用.....	242
CLAR 文件.....	243
生成清单文件.....	243
导入 CLAR 文件.....	243
导出 CLAR 文件.....	244
概念：CLAR 文件.....	244
JAXB 项目.....	245
创建 JAXB 项目.....	245
将 JAXB 项目添加为程序集项目中的前序项.....	246
概念：Workday Studio 中的 JAXB 支持.....	247
Workday Studio Ivy 插件.....	247
配置 Ivy 插件.....	247
在 Workday Studio 项目中添加 Workday Ivy 特性.....	247
解析具有 Ivy 定义的项目的依赖关系.....	248
向具有 Ivy 特性的项目添加从属 JAR.....	248

从 Workday Studio 项目中移除 Workday Ivy 特性.....	249
集成部署.....	249
添加连接.....	249
添加连接组.....	250
编辑连接.....	251
编辑连接组.....	251
移除连接.....	252
导入连接.....	252
导出连接.....	253
连接 Workday.....	253
将集成部署到 Workday.....	254
查看已部署到 Workday 的集成.....	255
配置 Studio 以使用 HTTPS 代理服务器.....	255
概念：Workday Studio 连接.....	256
概念：Cloud Explorer.....	256
参考信息：连接错误.....	256
Workday Studio 中的多重身份验证.....	257
在 Workday Studio 中启用多重身份验证.....	257
集成配置.....	259
将集成系统用户账户指定到集成.....	259
概念：集成系统安全性.....	259
集成启动.....	260
通过 Workday Studio 启动集成.....	260
复制启动 URL.....	260
创建启动配置.....	261
编辑启动配置.....	261
集成监控和问题排查.....	262
流程监视器.....	262
概念：集成事件编号.....	262
参考信息：流程监视器工具栏按钮.....	262
合并的报告查看器.....	263
加载有关集成的合并报告.....	263
将合并的报告保存到您的工作区.....	264
使用合并的报告查看 Workday 集成事件详细信息.....	264
使用合并的报告查看集成事件的请求消息.....	265
使用合并的报告查看器事件日志监控集成.....	266
使用合并的报告查看器配置文件日志监控集成.....	266
重新播放合并报告的程序集执行情况.....	267
将合并的报告审核事件与程序集元素相关联.....	267
在程序集图上显示合并的报告注释.....	268
概念：合并的报告查看器.....	268
参考信息：合并的报告错误消息.....	269
AUnit 程序集测试.....	272
为程序集项目创建 AUnit 测试框架.....	272
概念：AUnit 测试.....	273
参考信息：AUnit 测试注释.....	274
参考信息：AUnit 测试操作.....	276
Web 服务测试.....	276

使用 WSDL 文件测试 Web 服务.....	276
使用 XSD 文件测试 Web 服务.....	277
通过 Schema Explorer 测试单个 Web 服务.....	278
使用 Request Builder Wizard 测试 Get 操作和请求消息.....	279
配置 Web Service Tester 身份验证.....	279
配置 Web Service Tester X.509 身份验证.....	280
配置 Web Service Tester 高级 Web 服务安全性.....	281
概念：Web 服务测试.....	281
参考信息：Web Service Tester 工具栏按钮.....	281
程序集调试.....	283
调试程序集.....	283
概念：Assembly Debugger.....	284

Workday Studio 管理.....284

Workday Studio 首选项.....	284
概念：Assembly Editor 首选项.....	284
概念：连接首选项.....	286
概念：部署首选项.....	286
概念：HTTPS 首选项.....	286
概念：Ivy 首选项.....	286
概念：链接首选项.....	286
概念：MVEL 验证首选项.....	286
概念：封装集成首选项.....	286
概念：运行时首选项.....	287
概念：服务器首选项.....	287
概念：初始屏幕登录首选项.....	287
概念：状态栏首选项.....	287
概念：更新首选项.....	287
概念：XPath Explorer 首选项.....	287
参考信息：程序集调试运行时首选项.....	287
参考信息：程序集调试视图首选项.....	288
参考信息：Cloud Explorer 首选项.....	288
参考信息：合并的报告查看器首选项.....	288
参考信息：Java 项目设置首选项.....	289
参考信息：通知首选项.....	289
参考信息：流程监视器首选项.....	290
参考信息：Web Service Tester 首选项.....	290
Workday Studio 更新.....	290
更新 Workday Studio.....	290
Workday 程序集运行时变更.....	291
概念：程序集版本.....	291
参考信息：架构级别上的版本变更.....	291
参考信息：XML 级别上的版本变更.....	294
参考信息：Saxon XSLT 处理器变更.....	295

示例组合件.....297

示例：运行“Hello Cloud”示例程序集.....	297
示例：运行“Hello Workday Web Services”示例程序集.....	298
示例：运行常见使用模式示例程序集.....	298
示例：运行调试示例程序集.....	300
示例：运行 Workday-In 功能示例程序集.....	301
示例：运行错误处理示例程序集.....	303
示例：运行 AUnit 示例测试.....	303
示例：运行 FOPStep 示例程序集.....	304

- 示例：运行“Extract Organization Data for a Worker”示例程序集..... 305
- 示例：运行 PrismAnalytics 示例程序集..... 306
- 第三方薪资接口示例..... 308
 - 步骤：运行薪资接口示例..... 308
 - 创建薪资接口安全组..... 308
 - 创建 PayData Extract 集成系统..... 309
 - 测试 PayData Extract 集成系统..... 310
 - 部署 Workday Studio PayData 集成..... 311
 - 为 PayData 集成指定集成系统用户..... 312
 - 将 PayData Extract 集成与 PayData 集成相关联..... 312
 - 启动已关联的集成..... 313

设置时需考虑的事项：Workday Studio

注：此版块中描述的解决方案并非 Workday 服务的一部分。有关详细信息，请参见 [法律声明](#)。

在规划 Workday Studio 的配置和使用时，您可以利用本主题来帮助做出决策。本主题解释了以下内容：

- 设置目的。
- 如何与 Workday 的其他功能协同工作。
- 对下游的影响和产品间的交互。
- 安全性要求和业务流程配置。
- 实施前要考虑的问题和限制。

如需了解完整的配置详细信息，请参阅详细的任务说明。

用途

Workday Studio 是一种基于 Eclipse™ 的集成开发环境 (IDE)，使您能够执行以下操作：

- 构建自定义集成。
- 与外部端点交换批量数据。
- 将集成上传到 Workday。

Workday Studio 是外部软件应用程序，其安装与 Workday 的其余组件分开进行。

功能优势

通过 Workday Studio，您可以执行以下操作，从而提高工作效率：

- 创建集成，以支持比使用企业接口生成器 (EIB) 或连接器模板时更高的设计灵活性。
- 调试自定义集成的 XML 源代码。
- 通过 Workday 导出和导入大量数据。
- 将大型数据集转换为特定格式。示例：您可以将 CSV 数据集转换为 XML 格式。

用例

您需要：

- 无法通过 EIB 或连接器模板实现的自定义集成。示例：对数据集实施 XML 到文本 (XTT) 转换的集成。
- 入站集成，以便从多个数据源将批量数据导入到 Workday。
- 出站集成，以使用特定格式从 Workday 导出批量数据。
- 对自定义集成的 XML 源代码执行单元测试。

要考虑的问题

问题	需考虑的事项
您的组织是否与 Oracle 签订了长期支持协议？	如果已签订，您可以使用 Oracle JDK 1.8 安装 Workday Studio。 如果未签订，您可以使用 OpenJDK 8 安装 Workday Studio。
您使用哪个平台来安装 Workday Studio？	安装适合您的平台的安装文件： • MacOS (64 位) • Windows (32 位)

问题	需考虑的事项
	Windows (64 位)
您是否参加过 Studio 培训？	对于有意参加 Studio 培训的学员，Workday 提供了“使用 Workday Studio 创建集成”课程。进入培训目录了解更多信息。

建议

Workday 建议您先在预览沙盒环境中部署并测试自定义 Workday Studio 集成，然后再将这些集成转移到上线租户。

要求

Studio 集成需要与 Java 版本 8 兼容的 Java™ Development Kit (JDK)。此外，JDK 集成开发环境 (IDE) 需要与 Java 版本 17 兼容的 JDK。如果您：

- 已与 Oracle 签订许可协议，请使用 Oracle JDK 1.8 和 Oracle JDK 1.17。
- 未与 Oracle 签订协议，请使用 OpenJDK 8 和 OpenJDK 17。

两个 JDK 都是必需的。

限制

Workday Studio 集成输出文件的大小有以下限制：

- 单个文件不超过 1 GB (压缩后)。
- 所有文件总计不超过 3 GB (压缩后)。

租户设置

在 Workday 的“编辑租户设置 - 集成”任务中选中“需要源代码”复选框后，您即可访问 Workday Studio 集成的源代码。通过访问 Workday Studio 集成的源代码，您可以识别错误并提高效率。

安全性

建议在“Integration”功能区域中设置以下域。

域	需考虑的事项
Integration Build	将集成从 Workday Studio 部署到 Workday。
Integration Configure	在 Workday 中编辑 Workday Studio 集成。
Integration Debug	在 Workday 中调试 Workday Studio 集成。
Integration Event	在 Workday 中启动 Workday Studio 集成。
Integration Security	在 Workday 中为您的 Workday Studio 集成创建集成系统用户 (ISU) 账户。

业务流程

对于 Workday Studio 集成，您可以：

- 在“Integration”功能区域中配置“集成流程事件”业务流程，以控制集成的运行方式。
- 配置“集成”业务流程步骤，以在业务流程运行期间启动集成。示例：您可以向“填写 I-9 表”业务流程添加一个集成步骤。此步骤会在该业务流程中启动集成，以发送员工的数据。

报告

Workday 为 Workday Studio 集成提供以下报告：

报告	描述
集成事件	显示： - 正在进行或已完成的集成事件的汇总 - 每个集成事件的状态
View Integration System	显示您的 Workday 租户中可用的所有集成系统。

关系和关联点

Workday Studio 与 Workday 的不同部分进行交互，具体取决于集成的产品区域、目的和数据源：

- 入站集成会在 Workday 中添加和更新数据。
- 出站集成会从 Workday 中提取数据，但不会更改 Workday 中的数据。

Workday 提供了一个包含各种资源的 Workday Touchpoints Kit，可帮助您了解租户中的配置关系。如需进一步了解 [Workday Touchpoints Kit](#)，请访问 [Workday Community](#)。

Studio 基本功能

下载并安装 Windows 版 Workday Studio

关于此任务

Workday Studio 基于 Eclipse 开发平台构建。除平台和第三方前序项外，它还包含 150 多个 Eclipse 插件。使用前，您需要安装以下 2 个版本的 Java™ Development Kit (JDK)：

- 日食需要与 Java 版本 17 兼容的 JDK。
- Studio 运行时需要与 Java 版本 8 兼容的 JDK。

请安装这两个 JDK，然后确保：

- 将您的 Java 运行时环境设置为 Java 8。
- 您的 JDK 编译器合规级别设置为“Java 1.8”。

过程

1. 下载并安装与 Java 版本 17 兼容的 JDK。您可以从各种来源下载免费的开放源代码实施 OpenJDK。

示例：

- 选择适合您的操作系统的 [Zulu OpenJDK 17](#)。

注：

- Apple Slider 的用户必须使用 Apple Slider ARM JDK17。x86_64 JDK 将无法工作。
- Studio 需要专门与 Java 17 兼容的 JDK，而不是至少与 Java 17 兼容的 JDK。

2. 下载并安装与 Java 版本 8 兼容的 JDK。同样，您可以选择免费的开放源代码实施 OpenJDK。

示例：

- 选择适合您的操作系统的 [Zulu OpenJDK 8](#)

注：

- Apple Slider 的用户必须使用 Apple Slider ARM JDK8。x86_64 JDK 将无法工作。
- Studio 要求您的 JDK 特别与 Java 8 兼容，而不是至少与 Java 8 兼容。

3. 下载 Workday Studio for Windows [安装程序](#)。

4. 双击安装可执行文件来启动安装过程。若要将 Workday Studio 安装到 C:\Program Files 文件夹中，您必须以管理员身份运行该可执行文件，方法是右键单击可执行文件并选择“Run as Administrator”。

注：

您可以使用命令提示符应用程序来运行 Workday Studio 静默安装。在命令提示符中，导航到保存安装文件的目录并输入以下内容：

```
<Installation_Filename>.exe /S /J=<Java_Home> /D=<Installation_Directory>
```

其中，参数：

- /S 指定静默安装。
- /J 指定 JDK 17 路径。
- /RD 将安装 Workday 报告设计器。
- /D 指定安装文件夹（此参数必须是命令中的最后一个参数）。

示例：workday-studio-x86_64.exe /S /J=C:\Program Files\Zulu\zulu17.62.17-ca-jdk17.0.18-win_x64 /D=C:\Studio

5. 阅读许可协议，然后选中“接受条款”复选框。单击“下一步”。
6. 浏览到 JDK 17 的安装位置，然后单击“Next”。示例：个 JDK 安装位置：C:\Program Files\Zulu\zulu17.62.17-ca-jdk17.0.18-win_x64。
7. 如果 JVM 详细信息正确无误，请单击“确认”。否则，请单击“Back”。
8. 选择“Report Designer”包，将其与 Workday Studio 一起安装。您也可以稍后在 Studio 中安装报告设计器。
9. 浏览到要安装 Studio 的目录。单击“下一步”。
10. 当 Studio 显示“安装完成”消息时，单击“下一步”，然后单击“完成”。

注：运行 Java 的 日食 17 无法使用通过运行 Java 8 的日食创建的工作区。在启动时，您可以更新现有的工作区，使其与 Java 17 兼容。或者，您也可以创建一个全新的兼容工作区。更新工作区后，您将无法将其用于早期版本的 Studio。

11. 要确保 Java 8 可用作 Studio 中的 Java 运行时环境，请导航至其“首选项”面板：“Window” > “首选项” > “Java” > “Installed JRE”

如果缺少 Java 8，请单击“添加”，然后选择“标准虚拟机”作为“JRE 类型”。指定 JDK 8 的安装目录。

12. 确保选中“Java 8 JRE”旁边的复选框。

13. 检查“执行环境”是否完全匹配：“Window” > “Preferences” > “Java” > “Installed JREs” > “Execution Environments”

在“Execution Environments”面板中选择“JavaSE-1.8”，然后在“Compatible JREs”面板中选择所描述的“完全匹配”环境。

14. 在 Studio 的“首选项”面板中，导航到“Java” > “编译器”。

15. 在“JDK 合规性”下，从“编译器”合规级别 下拉菜单中选择“1.8”。

16. 单击“应用并关闭”。

注：在配置 Studio 之前，请参阅 [Workday 数据中心概览](#)。租户的数据中心页面包含的信息可供 IT 部门用于配置防火墙，并确保 Studio 能够正确访问 Workday。

下载并安装 macOS 版 Workday Studio

关于此任务

Workday Studio 基于 Eclipse 开发平台构建。除平台和第三方前序项外，它还包含 150 多个 Eclipse 插件。使用前，您需要安装以下 2 个版本的 Java™ Development Kit (JDK)：

- 日食需要与 Java 版本 17 兼容的 JDK。
- Studio 运行时需要与 Java 版本 8 兼容的 JDK。

请安装这两个 JDK，然后确保：

- 将您的 Java 运行时环境设置为 Java 8。
- 您的 JDK 编译器合规级别设置为“Java 1.8”。

过程

1. 下载并安装与 Java 版本 17 兼容的 JDK。您可以从各种来源下载免费的开放源代码实施 OpenJDK。

示例：

- 选择适合您的操作系统的 [Zulu OpenJDK 17](#)。

注：

- Apple Slider 的用户必须使用 Apple Slider ARM JDK17。x86_64 JDK 将无法工作。
- Studio 需要专门与 Java 17 兼容的 JDK，而不是至少与 Java 17 兼容的 JDK。

2. 下载并安装与 Java 版本 8 兼容的 JDK。同样，您可以选择免费的开放源代码实施 OpenJDK。

示例：

- 选择适合您的操作系统的 [Zulu OpenJDK 8](#)

注：

- Apple Slider 的用户必须使用 Apple Slider ARM JDK8。x86_64 JDK 将无法工作。
- Studio 要求您的 JDK 特别与 Java 8 兼容，而不是至少与 Java 8 兼容。

3. 下载正确的 Mac Studio 安装程序：

- [AppleSilicon](#)
- [x86, 64 位](#)

4. 要启动安装，请打开安装 .jar 文件。

5. 阅读许可协议，然后选中表示接受条款的复选框。单击“下一步”。

6. 浏览到 JDK 17 的安装位置，然后单击“Next”。示例：/Library/Java/JavaVirtualMachines/zulu-17.jdk/Contents/Home。

7. 如果 JVM 详细信息正确无误，请单击“确认”。否则，请单击“Back”。

8. 选择“Report Designer”包，将其与 Workday Studio 一起安装。您也可以稍后在 Studio 中安装报告设计器。

9. 浏览到要安装 Studio 的目录。单击“下一步”。

10. 当 Studio 显示“安装完成”消息时，单击“下一步”，然后单击“完成”。

注：运行 Java 的日食 17 无法使用通过运行 Java 8 的日食创建的工作区。在启动时，您可以更新现有的工作区，使其与 Java 17 兼容。或者，您也可以创建一个全新的兼容工作区。更新工作区后，您将无法将其用于早期版本的 Studio。

11. 要确保 Java 8 可用作 Studio 中的 Java 运行时环境，请导航至其“首选项”面板：“Workday Studio” > “首选项” > “Java” > “Installed JRE”。

如果缺少 Java 8，请单击“添加”，然后选择“标准虚拟机”作为“JRE 类型”。指定 JDK 8 的安装目录。

12. 确保选中“Java 8 JRE”旁边的复选框。

13. 检查“执行环境”是否完全匹配：“Workday Studio”>“首选项”>“Java”>“Installed JREs”>“Execution Environments”。

在“Execution Environments”面板中选择“JavaSE-1.8”，然后在“Compatible JREs”面板中选择所描述的“完全匹配”环境。

14. 在 Studio 的“首选项”面板中，导航到“Java”>“编译器”。

15. 在“JDK 合规性”下，从“编译器”合规级别 下拉菜单中选择“1.8”。

16. 单击“应用并关闭”。

注：在配置 Studio 之前，请参阅 [Workday 数据中心概览](#)。租户的数据中心页面包含的信息可供 IT 部门用于配置防火墙，并确保 Studio 能够正确访问 Workday。

概念：Workday Studio

Workday 提供了 2 种执行集成的方法：

- 连接器：连接到各种常见第三方服务的封装集成。
- 企业接口生成器 (EIB)：这是一种简单的工具，无需任何编程即可创建集成。

Workday Studio 是第三种选项，让您能够构建、拥有和支持更为复杂的集成，这些集成由 Workday 托管并代您运行。与连接器和企业接口生成器不同，Workday Studio 并不是 Workday 的一部分。它是一种基于 Eclipse 的集成开发环境 (IDE)，需要您单独下载。与租户中的其他任何集成一样，您可以启动和审核您使用 Workday Studio 构建的集成以及为其排定时间表。您可以通过 Workday Studio 本身或 Workday 应用程序启动通过 Workday Studio 创建的集成。

Workday Studio 的目标用户具有以下特点：

- 拥有开发技能。
- 熟悉集成编程概念。
- 熟练使用集成开发环境 (IDE)。

Workday Studio 平台有 4 个主要构成：

- Workday Studio IDE
- 集成、集合和 Cloud Repository
- Workday 应用程序
- Cloud Runtime

Workday Studio IDE

Workday Studio 是基于 Eclipse 的 IDE，可用来构建集成流，以便：

- 从 Workday 访问信息。
- 对其执行操作。
- (可选) 将其发送到某处。

这些流称为程序集。您可以在 Workday Studio Assembly Editor 中添加和连接名为构成和步骤的图形元素，以创建程序集。

Workday Studio 包含用于许多操作 (包括拆分、转换和路由) 的构成和步骤。Workday 特定构成具有内置的关于 Workday 如何表示信息的描述。

您可以先在 Workday Studio 中测试集成，然后再将集成直接部署到 Workday 租户。

Workday Studio 支持在程序集项目中使用 Java 扩展。您可提供自己的 Java 代码，并将其配置为 Spring Bean。您还可以将第三方 Java 库打包到集成中。

集成、集合和 Cloud Repository

Workday 使用 Cloud Repository 一词来表示其 Workday Studio 集成存储。在部署之前，Workday Studio 会将每个集成指定给一个集合。一个集合可以包含多个集成，一个集成可以属于多个集合。这意味着，您可以在不同集成和租户中轻松重复使用程序集和子程序集。

无论在任何时候，特定集合的实例都只能有一个在运行。如果您修改集合中的某个程序集项目并重新部署该项目，该项目将取代先前部署的任何版本。

Workday Studio 部署到 Workday Cloud Repository 的二进制文件称为 CLAR（全称为 CCloud ARchive，表示云归档）。一个 CLAR 包含以下内容：

- 所包含的集成项目中的所有工件，例如其 components.xml、XSLT 和文本架构文件。
- 开发者已添加到集成项目中的任何已编译 Java 代码或第三方 Java 库。

CLAR 是二进制对象，不包括源代码工件（例如 Java 源文件）。

Cloud Repository 以加密格式存储 CLAR。但是，请勿直接在集成中存储密码等信息。请在 Workday 应用程序中配置敏感详细信息，以动态管理这些信息。

Workday 应用程序

Workday 使用称为“集成系统”的抽象概念来对集成进行建模。在 Workday 中，对于每个集成，您可以指定：

- 哪些用户有权启动集成，以及在集成执行时集成拥有哪些权限。
- 在集成执行时如何以及何时向不同的 Workday 用户发出通知。

由于 Workday Studio 集成的设计和配置是分离的，因此您可以重新部署集成，而不会影响其配置。

Workday 允许在多种不同的情景中启动集成。您可以：

- 手动启动集成。
- 为集成排定时间表，使其作为事件自动运行。
- 选择在业务流程中自动启动集成。
- 使用 Web 服务客户端远程触发启动，以调用 Workday Web 服务。

您可以使用 Workday“流程监视器”报告来监控集成的执行情况。

Cloud Runtime

在集成启动时，Workday 会执行以下操作：

- 生成集成事件。
- 从 Cloud Repository 中检索集成。
- 指定一组称为 Cloud Runtime 的受保护处理资源。
- 在 Cloud Runtime 中执行集成。

如果 Workday 无法执行集成，则会更新 流程监视器，并将集成启动状态更新为 Failed。

如果 Workday 在执行过程中遇到问题（例如外部资源不可用），它将终止并更新 流程监视器，并将集成启动状态显示为 Completed with errors。

注：集成开发者有责任处理标记为“”的集成。Completed with errors。

Workday 会终止执行时间超过 2 小时的集成。开发者提供的 Java 代码通常无法读取或写入文件系统或 Java 系统属性。开发者提供的 Java 代码不能生成新 Java 线程。

安全性和隐私性

Workday Studio 直接为 2 种云环境提供支持：

- 沙盒
- 上线

您可以添加与其他 Workday 环境的连接。确保将正确的集成部署到适当的环境完全由您个人负责。标准 Workday Web 服务安全策略适用于使用 Workday Studio 集成访问的所有数据。

您可以在 Workday Studio 中离线创建程序集，但必须具有相应的安全配置才能将其与 Workday 集成。部署完毕后，集成将使用运行集成的用户的凭据和权限启动并运行。

Workday 并不了解您的集成是怎样的，也不了解其设计目的。确保集成按预期运行完全由您个人负责。您的集成与 Workday 租户的交互通过 Workday 公共 API 进行，因此您将继承所有相关的可靠性和安全性保证。

概念：集成开发生命周期

在以下不同的阶段中，您可以按需使用 Workday Studio 或 Workday 应用程序来开发和监控 Workday Studio 集成：

阶段	描述
集成设计	在 Workday Studio 中，开发构成集成的程序集。（可选）创建程序集模块，以便重复使用和共享集成模式和配置。 每个程序集至少有 1 个“workday-in”传输，用来表示执行的起点。您可选择为每个“workday-in”传输定义以下内容： - 启动参数，这些参数使客户端能够将信息作为输入传递到集成中。示例：薪资集成可能会指定它需要 2 个启动参数：payPeriod 与 payGroup。这使集成的用户能够指定在计算薪资时要使用的月份和员工组。 - 程序集在运行时使用的 Workday 服务。Workday 为属性映射、序列生成、事务日志记录、文档检索和传送提供了通用集成服务。它还提供侦听器服务、报告服务及自定义对象服务。
集成部署	在 Workday Studio 中，连接到 Workday Cloud Repository，以部署集成。对于集成中定义的每个“workday-in”传输，Workday 会注册一个称为集成系统的抽象实体。集成系统中包括所有必要的启动参数和服务。
集成配置	在 Workday 中，您可以配置集成的某些方面，包括权限和通知。
集成启动	通过集成系统，Workday 可以启动和跟踪您的集成。但为了方便开发和测试，您也可以从 Workday Studio 进行启动。
集成监控和问题排查	在 Workday Studio 或 Workday 中跟踪集成进度。在 Workday Studio 中，您可以在“Process Monitor”中监控集成的执行情况。您还可以在集成流程完成后查看合并的报告。

概念：MediationContext 对象

在程序集执行期间，Workday 会创建一个名为“”的临时所有者对象。MediationContext。该 MediationContext 包含 3 种主要类型的信息：

- **MediationMessage**：一个对象，其中包含 Studio 使用输出传输器最终发送到外部目标的文本或二进制数据消息。示例：HTTP 输出、FTP 输出。
- **变量**：消息数据的临时容器。
- **属性**：由名称-对象对组成的映射，集合元素可在非消息状态下存储暂时性数据。

程序集中的所有处理都涉及从中读取或写入 `MediationContext`。

注：当您访问 `MediationContext` 对象（使用 MVEL），则访问者为 `context`。示

例：`context.getErrorMessage()`。在 Java 中是 `mediationContext`。示例：`mediationContext.getErrorMessage()`。

概念：集成系统和事件

集成系统

在 Workday 中，集成系统是指对外部系统或 Workday 端点进行集成的抽象表示。集成系统提供 Workday 用来启动和跟踪集成的框架。您可以使用集成系统来配置以下内容：

- **安全性**：指定谁可以启动集成，以及集成执行时的权限。
- **通知**：指定谁会在集成执行时收到通知。
- **参数**：您可以设置参数来调用集成。示例：您可以配置集成所需的端点和用户凭据，以将数据上传到外部 SFTP 网站。
- **触发器**：您可以将集成配置为通过以下方式触发：
 - 手动创建。
 - 作为订阅的一部分。
 - 使用 API。
 - 按时间表。
 - 作为业务流程的一部分。

Workday 使用集成系统统一表示所有集成（无论其来源如何）。

在使用 Workday Studio 开发集成并将其部署到 Workday 时，您会自动为该集成中的每个“workday-in”传输注册新的集成系统。

集成事件

集成启动时，Workday 会创建一个名为“集成事件”的对象。此对象对于每次调用都是唯一的，并包含一个名为“集成事件编号”的编号。Workday 使用此唯一编号来跟踪所有正在执行的集成。您可以使用 Workday 中的“Process Monitor”来查看集成事件。

在创建集成事件对象的同时，Workday 还会创建集成事件请求消息，并将其发送至集成。此请求包含有关集成系统和集成事件编号的信息。另外，其中还包含启动时提供的所有参数值。然后，在向 Workday Web 服务请求信息时，集成可以使用这些信息。集成还可以使用这些信息将更新信息发送回 Workday。示例：有一条状态消息，告知用户已成功处理 100 条员工记录。Workday 会将这些更新添加到集成事件对象中。

参考信息：界面基础知识

Workday Studio 界面由多个视图组成。要显示或隐藏视图，请选择“Window” > “Show View” > “Other...”，然后展开“Workday”文件夹。

Workday Studio 中最常用的视图包括：

查看	描述
Project Explorer	Workday Studio 的项目导航器。显示当前工作区中保存的项目。每个文件夹都包含该项目的有关文件，并且引用该项目所属的集合。
Cloud Explorer	显示 Workday Studio 可以部署到的集成云服务器，以及当前部署的集合。您可以添加与多个环境的连接。如果出现连接、身份验证或授权错误，Workday Studio 会在连接图标上显示错误或警告符号。
Palette	显示您可以添加的所有程序集构成。具有 3 个选项卡式子视图： • Design：显示程序集图。 • Source：显示 XML 源。 • Message Builder：一个图形视图，可简化为程序集元素（例如“write”步骤）定义消息内容的过程。
Assembly Editor	程序集的图形表示。要添加新的构成和步骤，请从“Palette”中拖放构成和步骤。要连接构成，请将光标悬停在源构成上以显示连接图柄，然后将图柄拖动到目标构成。
Properties	显示程序集图中当前选定的程序集元素的配置详细信息。使用视图左侧显示的选项卡编辑属性和参数。
流程监视器	在您启动集成时显示有关程序集流程的信息。选择一个流程可在“Process Details”视图中显示其详细信息。
Schema Explorer	让您能够浏览 WSDL 和 XML 架构文件的内容，并提供多项实用程序功能，以便您开发与 Web 服务交互的程序集。
Consolidated Report Viewer	让您能够分析集成中的程序集处理步骤。

集成设计

程序集

创建程序集项目

关于此任务

程序集定义了基于消息的集成流程，其中包括注释、传输、中介组件、中介步骤和错误处理程序。

过程

1. 选择“文件”>“新建”>“Workday 程序集项目”。
2. 在“项目名称”字段中，输入项目的名称。
项目名称不能包含空格。

3. 从“Target Runtime”列表中，选择“Workday Runtime”。
4. 从“Configurations”列表中，选择以下配置选项之一：

选项	描述
Assembly	创建简单的程序集项目。
Assembly (with Java support)	创建包含 Java 分面的项目。
自定义	允许您为程序集项目指定项目分面。

5. 单击“下一步”。输入一个集合名称。
“Collection name”字段会自动填充为您的项目名称。
6. 单击“完成”以创建项目。

注：如果您在工作区之外创建了程序集项目，则可以在 Studio 中为该项目指定新名称，而无需更改原始位置中的文件夹名称。这也适用于您从工作区外部导入的项目。为此，请选择“文件”>“重命名”。

结果

新的程序集项目将显示在“项目浏览器”视图中。其中包含一个 assembly.xml 文件，您现在可以在 Assembly Editor 中打开该文件。

下一步做什么

使用 Assembly Editor，您可以通过添加和关联传输、中介构成和中介步骤来定义集成。

向构成添加步骤

开始之前

创建一个程序集项目。

关于此任务

当消息通过集成时，步骤会对其执行各项操作。您只能向“async-mediation”和“sync-mediation”构成添加步骤。

过程

1. 在 Assembly Editor 中打开程序集。
2. 将“async-mediation”或“sync-mediation”构成添加到程序集。
3. 将一个步骤从“Palette”拖动到您的“async-mediation”或“sync-mediation”构成。按执行顺序排列步骤。完成此任务时，请参考以下信息：

Palette 种类	描述
Transform	显示中介步骤，这些步骤使您能够在消息通过您的集成时对其进行转换。
Logging & Eventing	显示中介步骤，这些步骤使您能够在消息通过您的集成时对其进行记录。
Message Manipulation	显示中介步骤，这些步骤使您能够在消息通过您的集成时对其进行操作。
验证	显示中介步骤，这些步骤使您能够在消息通过您的集成时对其进行验证。
Encoding	显示中介步骤，这些步骤使您能够在消息通过您的集成时对其进行编码。

Palette 种类	描述
其他	显示其他中介步骤，这些步骤使您能够在消息通过集成时对其执行各项操作。

- 单击“保存”。

结果

Workday Studio 会在程序集图中显示构成内的新步骤。

下一步做什么

配置新步骤的属性。

添加子元素

开始之前

创建一个程序集项目。

关于此任务

某些程序集元素支持子元素。子元素是可选元素。

过程

- 在 Assembly Editor 中打开程序集。
- 在程序集图中，添加一个支持子元素的程序集元素。
- 将光标悬停在该程序集元素的标题上，以显示其子元素选项。
- 单击某个子元素选项，以将其添加到程序集元素中。

结果

Workday Studio 会在程序集元素中显示子元素。

下一步做什么

配置子元素的属性。

连接程序集元素

开始之前

创建一个程序集项目。

关于此任务

程序集元素之间的连接定义了消息在程序集中的传递路径。在 Assembly Editor 中，您可以使用“Outgoing connection”图柄（蓝色方块）或“Incoming connection”图柄（橙色方块）来连接程序集元素。

过程

- 在 Assembly Editor 中打开程序集。
- 向程序集图添加 2 个程序集元素。
- 将光标放在第一个程序集元素的标题上，以显示其连接图柄。将“Outgoing connection”图柄（蓝色方块）从第一个程序集元素拖动到第二个程序集元素。

4. 或者，将光标放在第二个程序集元素的标题上，然后将“Incoming connection”图柄（橙色方块）拖动到第一个程序集元素上。

结果

第一个程序集元素的消息输出将成为第二个程序集元素的输入。

配置程序集元素属性

开始之前

向程序集图添加程序集元素。

关于此任务

Workday Studio 的“Palette”中的每个元素都具有“Properties”视图，您可以在其中配置其功能。

过程

- 1. 右键单击一个程序集元素，选择“Show Properties View”。
- 2. 对于必需的属性，其图标的左上角带有一个星号。完成此任务时，请参考以下信息：

属性图标	属性描述
	在程序集中必须是唯一的值。
	程序集元素的输入。
	程序集的输出。
	MVEL 表达式。
	MVEL 模板。
	布尔值。
	选项。
	数值。
	属性值。
	Java 类实现。
	引用 Java 类实现的 Spring Bean。

注：如果属性支持 MVEL，则相关 MVEL 图标会添加到属性自身的图标上。示例：

- 3. 要比较属性视图，请执行以下操作：
 - a) 在程序集图中选择一个构成。
 - b) 按住 Ctrl 键，同时选择第二个构成。
 - c) 右键单击两个构成之一，然后选择“Compare Properties”。第一个构成的“Properties”视图显示在左侧。第二个构成的“Properties”视图显示在右侧，并固定在该构成上。当您选择其他构成时，左侧的视图会改变。
- 4. 在支持 MVEL 的属性字段中，同时按 Ctrl 和空格键可访问内容助手，它会提供一组上下文相关的代码完成选项。

注：内容助手仅在包含 Java 分面的程序集项目中可用。

管理集合和项目

开始之前

创建一个程序集项目。

关于此任务

在 Workday Studio 中，您可以使用集合来封装和部署多个项目。一个项目始终属于至少一个集合，并且可以同时作为多个集合的成员。

过程

- 1. 单击“Workday” > “管理收款”。
- 如果您在左侧窗格中展开集合，Workday 会显示它包含的项目。不属于该集合的项目会显示在右侧窗格中。
- 2. 完成此任务时，请参考以下信息：

按钮	描述
New Collection	向您的工作区添加新集合。
Rename Collection	为选定集合重命名。
Remove Collection	从工作区中移除选定集合。
Add to Collection	向集合添加选定项目。
Add All Integrations to the Collection	将所有项目添加到选定集合。
Remove the Integration from Collection	从集合中移除选定项目。

注：确保集合中的所有项目都具有相同的程序集版本。否则，Workday Studio 会显示错误标记。您可以在 Assembly Editor 的“Properties”视图中为程序集配置“Version”属性值。

- 3. 单击“Finish”。

声明命名空间前缀

开始之前

创建一个程序集项目。

关于此任务

在 Workday Studio 中，命名空间前缀是带有预定义 URI 值的缩写。您可以声明命名空间前缀，以引用程序集中的特定 URI 值。

过程

1. 在 Assembly Editor 中打开一个程序集。
2. 右键单击程序集图的背景，然后选择“Show Properties View”。
3. 单击“Namespace Prefixes”选项卡。
4. 单击“Add”以添加新的命名空间前缀。完成此任务时，请参考以下信息：

选项	描述
前缀	在此处输入前缀名称。
Namespace	在此处输入命名空间 URL。

5. 要编辑新的命名空间前缀，请单击“Edit”。
6. 要移除新的命名空间前缀，请单击“Remove”。

结果

程序集的命名空间前缀将显示在 XML 架构中。

创建路径变量

开始之前

创建一个程序集项目。

关于此任务

路径变量指定资源文件在工作区文件夹中的位置。在程序集项目中关联平台资源时，您可以使用工作区路径变量。

过程

1. 选择“Windows” > “首选项” > “常规” > “工作区” > “链接的资源”。
 2. 单击“新建”。
- 完成此任务时，请参考以下信息：

选项	描述
名称	输入路径变量的名称。
地点	输入路径变量的文件位置。

3. 单击“确定”。

结果

现在，您的新路径变量会指定资源文件在工作区文件夹中的位置。

概念：程序集 **Palette**

“Palette”显示 Workday Studio 中可用的所有程序集元素。要向程序集图添加某个程序集元素，请将该元素从“Palette”拖动到 Assembly Editor。“Palette”中列出了以下种类的程序集元素：

Palette 种类	描述
Favorites	显示您收藏的程序集元素。要向此种类添加程序集元素，您可单击这些元素旁边的空心星形图标。要将元素从此种类中移除，您可单击相应元素旁边的高亮星形图标。

Palette 种类	描述
Annotation	显示注释，这些注释使您能够对程序集图进行构建、整理和标注。
Module Library	显示模块程序集元素，这些元素使您能够为集成配置 Workday Web 服务调用和全局错误处理程序。
In Transports	显示从 Workday 或本地资源接收消息的 in-transport 程序集元素。
Out Transports	显示向资源发送消息的 out-transport 程序集元素。
Components	显示中介构成，这些构成使您能够在消息通过您的集成时对其进行操作和转换。
Common Components	显示可用于构建集成的预封装子程序集。
Workspace Components	显示您已创建且保存到工作区的自定义子程序集。
Error Handlers	显示使您能够捕获集成中的错误的错误处理程序。
Transform	显示中介步骤，这些步骤使您能够在消息通过您的集成时对其进行转换。
Logging & Eventing	显示中介步骤，这些步骤使您能够在消息通过您的集成时对其进行记录。
Message Manipulation	显示中介步骤，这些步骤使您能够在消息通过您的集成时对其进行操作。
验证	显示中介步骤，这些步骤使您能够在消息通过您的集成时对其进行验证。
Encoding	显示中介步骤，这些步骤使您能够在消息通过您的集成时对其进行编码。
其他	显示中介步骤，这些步骤使您能够在消息通过您的集成时对其执行各项操作。
联系人	显示“Route To”和“Route Response To”连接，这些连接使您能够定义消息在集成中的传递路径。

要手动筛选“Palette”中显示的程序集元素，请使用“type filter text”字段。Workday Studio 还会直接在“type filter text”字段下方提供以下预配置的种类筛选器：

种类筛选器	描述
Restore	清除之前的筛选器选择。
Transports	仅显示传输的“Palette”种类。
Components	仅显示中介构成的“Palette”种类。
步骤	仅显示中介步骤的“Palette”种类。
Handlers and Connections	仅显示错误处理程序和连接的“Palette”种类。
Filter Categories	显示“Palette Category Filter”窗口。
Expand All	展开所有“Palette”种类。
Collapse All	收起所有“Palette”种类。

Workday Studio 会自动在 Assembly Editor 旁边显示“Palette”。要在单独的视图中显示“Palette”，请选择“Window” > “Show View” > “Other” > “General” > “Palette”，然后单击“OK”。您还可以右键单击“Palette”并选择“Layout”来更改“Palette”的布局。

relatedlinks
参考信息

[Workday 31 What's New Post: Workday Studio](#)

概念：构成和步骤

您只能向“async-mediation”和“sync-mediation”构成添加步骤。

Workday Studio 提供了支持转换、验证和日志记录等操作的步骤。您还可以使用 Spring Bean 编写自定义步骤。

“async-mediation”构成支持 1 个处理链，仅可处理请求消息。“sync-mediation”构成支持 2 个处理链，可处理请求消息和请求-响应消息。

步骤具有“Input”和“Output”属性，分别供步骤获取作为输入的数据和指明输出方向。以下是 5 个可能的值：

选项	描述
message	整个消息，包括 rootpart 和附件。如果您指定此选项，则 Workday Studio 会创建一条新消息，从而有效移除所有附件。
soapbody	消息 rootpart 中 SOAP 主体的内容。
attachment	消息的 MIME 附件。指定要访问的附件的索引值。您还可以通过将索引值指定为 -1，来向消息添加新附件。
rootpart	MIME 消息 rootpart 的内容，如果消息不是 MIME，则为消息本身。如果为输出属性指定此值，则会保留原始消息中的所有附件。
变量	中的变量 MediationContext。指定要使用的变量的名称。

概念：程序集注释

通过注释，您可以对程序集图进行构建、整理和标注。Workday Studio 在“Palette”的“Annotations”版块下提供以下选项：

- Rectangle
- Ellipse
- Note
- 文本
- Swimlane

您可以：

- 使用“Rectangle”“Ellipse”“Note”“Text”注释，为程序集添加常规解释性文本。
- 使用“Show/Hide annotations”按钮，显示或隐藏“Rectangle”“Ellipse”“Note”“Text”注释。仅当“Show annotations”处于有效状态时，您才能向程序集图添加新注释。
- 使用“Rectangle”“Ellipse”“Note”“Text”注释上的图钉图标，将这些注释选项固定到程序集。已固定的注释会始终显示在图表上，即使在“Hide annotations”处于有效状态时也不例外。

泳道

通过泳道，您可以将程序集构成分组，以便进行组织整理。

您可以：

- 单击“Show/Hide swimlane descriptions”按钮，以显示或隐藏泳道描述。
- 单击“Expand all swimlanes”和“Collapse all swimlanes”按钮，以展开和折叠所有泳道。
- 单击各个泳道标头中的加号和减号图标，以展开和折叠各个泳道。泳道状态具有永久性。在您打开程序集时，泳道会以其在保存程序集时的收起或可见状态显示。

泳道支持嵌套。

relatedlinks

参考信息

[Workday 31 What's New Post: Workday Studio](#)

概念：程序集项目的链接资源文件

链接的资源是指存储在工作区中的文件，您可以在多个程序集项目中链接到这些文件。

要查看某个项目的链接资源列表，请在“Project Explorer”视图中右键单击该项目。选择“Properties” > “Workday” > “Linked Resources”，以显示“Linked Resources”窗口。

要从项目中移除链接的资源文件，请在“Project Explorer”视图中右键单击该资源文件，然后选择“Remove linked resource”。

要将链接的资源复制到项目中，请在“Project Explorer”视图中右键单击该资源文件，然后选择“Convert linked resource to copy”。

概念：“Quick Outline”视图

利用“Quick Outline”视图，您可以轻松找到程序集的各个元素。要访问“Quick Outline”视图，请在“Assembly Editor”获得焦点的情况下按 Ctrl+O。

注：您无法通过“Window” > “Show View”菜单访问“Quick Outline”视图。

“Quick Outline”视图允许您按以下类型搜索程序集：

搜索类型	描述
MVEL Expression	搜索 MVEL 表达式和模板属性。使用此视图可搜索文本值的实例。
编号	搜索程序集构成编号。
可变	搜索变量名称。Workday Studio 将变量分为两类：为访问步骤定义的变量，以及为“write”步骤定义的变量。

要选择搜索类型，请单击视图顶部搜索字段下的相关按钮。

要进行搜索，请在搜索字段中输入文本。当您在搜索结果列表中选择 MVEL 表达式、构成编号或变量时，Workday Studio 会跳转到程序集中的相关构成。

参考信息：程序集属性

属性	XML 属性名称	描述
编号	id	程序集的唯一编号。
版本	version	程序集版本号。
审核日志	audit-log	程序集的消息日志设置。选项包括：

属性	XML 属性名称	描述
		• True：启用并存储所有日志。 • False：禁用所有日志。 • On Error：启用所有日志，且仅存储错误日志。
回调处理程序 bean	callback-handler-bean	为程序集处理异步通信的 Spingle bean。
集成模块	beans	程序集中的 Loading 模块。
命名空间前缀	xmlns	程序集中的命名空间前缀。

参考信息：泳道属性

“Common”选项卡

属性	描述
名称	泳道的名称。显示在泳道的标头中。
描述	泳道的描述。此属性会在您选择了“Show swimlane descriptions”时显示。
入职简介	泳道的方向。选项包括： • HORIZONTAL • VERTICAL
目标协调	泳道中各个构成的对齐方式。选项包括： • BEGINNING • MIDDLE • END
Label Alignment	泳道标签的对齐方式。选项包括： • LEFT • CENTER • RIGHT
Top border color	泳道顶部边框的颜色。
Line color	泳道侧面和底部边框的颜色。
Fill color	泳道的背景颜色。
Font	泳道标签的字体。
Line style	泳道侧面和底部边框的样式。选项包括： • SOLID • DASH • DOT • DASHDOT • DASHDOTDOT

参考信息：连接线属性

“Common”选项卡

属性	描述
行类型	程序集图中用于显示“Route To”和“Route Response To”连接的线的类型。选项包括： • straight • curved

程序集传输

在 Workday Studio 中将枚举用作启动参数

开始之前
创建一个包含“workday-in”传输的程序集项目。

关于此任务
在 Workday Studio 中，您可以将枚举用作集成启动参数。这会限制您在启动集成时可在“Integration Criteria”选项卡中选择的值。您可以在 Workday Studio 中定义枚举，也可以引用租户中的现有枚举定义。

- 过程
1. 在 Assembly Editor 中打开程序集。
 2. 在“workday-in”传输的“Properties”视图的“Launch Parameters”选项卡上，单击“Add Enumeration Type Parameter”。
 3. 单击“Edit Type”。
 4. 配置枚举时，请参考以下信息：

选项	描述
Enumeration Reference	为租户中的现有枚举定义输入类型名称。
Enumeration Definition	为枚举定义输入唯一的类型名称。 注：当您为现有枚举定义输入类型名称时，Workday 会重设该定义。 添加枚举允许的值。

下一步做什么
部署您的集成。
relatedlinks
任务
[将集成部署到 Workday on page 254](#)

概念：程序集传输

- Workday Studio 在“Palette”上提供了以下两类程序集传输：
- In Transports
 - Out Transports

入站传输接收来自 Workday 或本地资源的消息，然后将这些消息转发给程序集中定义的其他元素进行处理。入站传输始终表示处理链的开始。

out-transport 与 HTTP 端点、Workday 应用程序或 FTP 站点等资源交换消息。包含“local-out”传输的程序集可以与包含“local-in”传输的子程序集通信。

Workday Studio 在“Palette”中支持以下入站传输程序集元素：

名称	函数	注释
workday-in	定义程序集的执行起始点。	• 侦听 Workday 应用程序请求消息。 • 支持集成的启动参数和服务定义。 • 同步（请求-响应）。
local-in	定义程序集或子程序集的入口点。	• 与包含“local-out”传输的程序集交换消息。 • 支持参数和输出参数。 • 同步（请求-响应）。

Workday Studio 在“Palette”中支持以下 out-transport 程序集元素：

名称	函数	注释
workday-out-rest	将 Workday 应用程序 REST 请求消息发送到 HTTP URL。	• 同步（请求-响应）。
workday-out-soap	将 Workday 应用程序 SOAP 请求消息发送到 HTTP URL。	• 同步（请求-响应）。
local-out	与包含“local-in”传输的程序集和子程序集交换消息。	• 同步（请求-响应）。
http-out	将 HTTP 请求消息发送到 HTTP URL。	• 支持以下 HTTP 子元素： • http-basic-auth • http-custom-auth • https-properties • 同步（请求-响应）。
ftp-out	将消息发送到 FTP 服务器。	• 异步（仅请求）。
ftps-out	使用 FTP over SSL (FTPS) 协议将消息安全地发送到 FTP 服务器。	• 异步（仅请求）。
sftp-out	使用 FTP over SSH (SFTP) 协议将消息安全地发送到 FTP 服务器。	• 异步（仅请求）。
xmpp-out	将文本消息写入 Jabber 服务器。示例：Google Gmail 服务器。	• 异步（仅请求）。
custom-out	创建可自定义的 out-transport。	• 同步性由 Spring Bean 实现来定义。

名称	函数	注释
email-out	将电子邮件写入 SMTP (邮件) 服务器。	- Workday Studio 会检查以下属性中是否存在前导空格和尾随空格： - 如要执行以下操作 - 主题 - 自 - Bcc - Cc - Host - Port - User - Password - 回复到 - 异步 (仅请求)。

概念：枚举

枚举会将属性值限定为一组字面值，例如成本中心代码。在设计集成时，您可以使用枚举来定义一组常量，以用作字面值。这样，您就可以限制集成所用数据的范围。

您可以使用以下任务在 Workday 中创建和维护枚举定义：

- Create Integration Enumeration Definition
- Edit Integration Enumeration Definition
- Delete Integration Enumeration Definition
- View Integration Enumeration Definition

您可以在以下元素中使用枚举：

- 启动参数
- 集成属性
- 集成映射

您可通过以下方式指定枚举：

- 在 Workday Studio 中指定。
- 使用 Workday 中的“Create Integration Generic Service”任务指定。

您可以使用以下某种方法在 Workday Studio 集成中指定枚举：

- 在 Workday Studio 中创建枚举定义，并在启动 Workday Studio 集成时将其部署到您的租户。
- 创建一个枚举引用，引用租户中的现有枚举定义。

您可以在多个集成中使用相同的枚举定义，方法是：选择 1 个集成来提供定义，并在其他集成中引用该定义。

relatedlinks

任务

[创建属性映射服务](#) on page 233

参考信息：Workday-In 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集内 Workday-in 传输的唯一编号。
寄送目的地	routes-to	Workday 传输将消息定向到的组合元素。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	按程序集指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
集成系统名称	integration-system-name	您的集成系统的名称。

“Launch Parameters”选项卡

启动参数	描述
Simple Type Parameter	使用字符串字面值的启动参数。
Reference Type Parameter	使用对 Workday 类报告字段 (CRF) 和 Web 服务的引用的启动参数。
Enumeration Type Parameter	使用 Workday 类报告字段和枚举定义的启动参数。

“Services”选项卡

集成服务	描述
Attribute Map Service	集成中所用的集成属性和集成映射。
Delivery Service	将文档从 Workday 租户传送到目标端点。
Listener Service	使集成能够通过您的 Workday 租户接收来自第三方客户端的启动消息。
Retrieval Service	从远程客户端检索文档，并将其存储到您的 Workday 租户中。
Sequence Generator Service	每次运行集成时，系统都会为文档生成唯一的有序文件名。

集成服务	描述
Transaction Log Service	显示集成系统在 Workday 租户中订阅的事务事件的日志。
Report Service	为您在集成中引用的任何自定义报告指定别名。
Custom Object Service	为您在集成中引用的任何自定义对象指定别名。
Service Reference	引用在同一集合内的其他“workday-in”传输中配置的任何集成服务。

参考信息：Local-In 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中 local-in 传输的唯一编号。
寄送目的地	routes-to	local-in 传输将消息定向到的组合元素。
访问权限	access	指定您的子程序集是“公有”还是“私有”。 “Palette”仅显示“Workspace Components”种类中的公开子程序集。
使用全局错误处理程序	use-global-error-handlers	确定是否在子装配层级处理子装配内发生的错误。默认值为 false，表示错误由调用了子部件的部件集进行处理。
图标	icon	为子部件指定自定义图标。支持 24x24 像素的 PNG 或 GIF 图标。
工具提示	tooltip	子装配的工具提示。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	存储消息 不适用于 本地输入 传输。对于“local-in”传输，Workday Studio 会忽略此设置。
传输类	transport-class	按程序集指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。

“参数”选项卡

local-in 传输的参数可以指定任何值。MediationContext 您可以设置的属性。

属性	XML 属性名称	描述
名称	name	local-in 参数的名称。
说明文档	documentation	local-in 参数的文本描述。
必需	required	确定 本地转入 传输是否需要您的参数。
类型	type	指定您的参数类型。选项包括： • string • integer • long • double • float • 布尔值
默认值	default	参数的默认值。
验证	validation	布尔值 MVEL 表达式，其计算结果必须为 true，否则 Studio 会引发错误。使子装配件开发人员能够对参数值施加约束。

“转出参数”选项卡

local-in 传输的 out-参数可以指定任何值。MediationContext 子装配件检索的属性。

属性	XML 属性名称	描述
名称	name	您的转出参数的名称。
说明文档	documentation	转出参数的文本描述。

参考信息：Workday-Out-Rest 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集内的 Workday-out-REST 交通的唯一编号。
将回复发送至	routes-response-to	Workday-out-REST 传输的响应消息目标。
执行时间	execute-when	一个条件，用于确定是否执行 Workday-out-REST 传输。
额外路径	extra-path	指定 Workday 租户中 Studio 发送 REST 请求时使用的路径。Workday Studio 会自动为此路径设置并附加 Workday 服务端点。示例：以下路径中的端点从名为 ACME 的租户中调

属性A	XML 属性名称	描述
		用 Workday 业务网站目录 报告（以 JSON 格式请求）： systemreport2/ACME/ Business_Site_Directory? format=json

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	按程序集指定一个备选的实现类。 您可以通过编辑 WEB-INF/ classes/spring/assembly- bean.xml 文件来全局更改实现类。
方法	method	使您能够向 Workday-out-REST 传输添加 REST 方法。示例： put，get，post。
失败消息	failure-message	一条自定义消息，用于替换在您调用 Workday-out-REST 端点时返回的任何异常消息。

参考信息：Workday-Out-Soap 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集内的 workday-out-soap 传输的唯一编号。
将回复发送至	routes-response-to	Workday-out-soap 传输的响应消息的目标。
执行时间	execute-when	确定是否执行 Workday-out-soap 传输的条件。
应用程序	application	传输将内容发送到的 Workday 应用程序的名称。Workday Studio 使用您在此字段中输入的名称来创建 URI，以指定 Workday 应用程序的位置。
版本	version	SOAP 请求调用的 Web 服务版本。 注：当您在 XSLT 请求中指定 Web 服务版本时，该版本优先。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
失败消息	failure-message	一条自定义消息，用于替换在调用 Workday-out-soap 端点时返回的任何异常消息。
替换为 Soap 错误	replace-with-soap-fault	用 SOAP 错误替换失败的 Workday-out-soap 消息。SOAP 错误包含已提取的 faultcode 与 faultstring 属性值。

参考信息：Local-Out 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中 本地输出 传输的唯一编号。
将回复发送至	routes-response-to	本地出 站传输的响应消息的目标。
执行时间	execute-when	确定是否执行 本地输出 传输的条件。
端点	endpoint	本地输出 传输将消息定向到的目标端点地址。 端点地址遵循以下格式 application-name/component-ID，其中： - application-name 是包含目标程序集的项目的名称。 - component-ID 是目标组件的编号。 示例：<cc:local-out id="local" endpoint="myassembly/mediation1"/>

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	clone-request	指定是否为 本地输出 传输协议创建请求消息的副本，以便对消息进行的任何子程序集修改都不会影响其他子程序集。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配件在标记 MediationContext 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

属性	XML 属性名称	描述
名称	name	参数子元素的名称。
值	value	参数子元素的值。

参考信息：HTTP-Out 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中 http 输出 传输协议的唯一编号。

属性A	XML 属性名称	描述
将回复发送至	routes-response-to	http-out 传输协议的响应消息的目标。 注: 使用 http 输出 传输协议填充 MediationContext 属性 http.response.status 以及远程服务器返回的标准 HTTP 响应代码。示例: 202 (已接受)、403 (禁止)。程序集可以使用此属性中的值来确定对不同响应的处理方式。
执行时间	execute-when	确定是否执行 http 输出 传输的条件。
端点	endpoint	http-out 传输将请求消息定向到的目标 HTTP URL 地址。 示例: http://myhost.example.org/MyService
HTTP 方法	http-method	http-out 传输协议支持的 HTTP 请求方法。示例: DELETE, GET, POST, PUT 和 PATCH。 注: 如要使用 PATCH 方法, 则必须设置 wd.http.client 属性 apache。如果您不希望下游传输也使用 HTTP 客户端, 请重置 wd.http.client 属性 null。 如果您没有为此属性指定方法, 则 Studio 会使用 POST 方法。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知, 否则 无 指定内容。 要存储消息, 请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要, 为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值, 还是使用它来重设传输协议的 Endpoint 属性中的值。

属性A	XML 属性名称	描述
		使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
连接超时	connect-timeout	以毫秒为单位，指定与 HTTP 服务器建立连接的超时期限。如果您未指定值，会话将在 300000 毫秒（5 分钟）后超时。
响应超时	response-timeout	以毫秒为单位，指定等待 HTTP 服务器响应的超时期限。如果您未指定值，会话将在 21600000 毫秒（6 小时）后超时。 如果 HTTP 连接处于无响应状态的时长超过此时限，“http-out”传输将失败。
回复时显示错误	error-as-response	让您能够将服务器的错误响应消息复制到 http 输出 传输协议上的消息。
关闭连接	close-connection	添加一个 Connection: close http-out 传输协议报文标头。
流式传输	streaming	启用 HTTP 流式传输，请使用 PUT 与 POST HTTP 方法
Auto Inproc	auto-inproc	指定是否对 http 输出的 请求消息使用进程内传输通信。这使您能够直接向位于同一服务器内的程序集端点发送消息，而无需使用本地主机的网络 I/O 功能。
重试	retries	指定 http-out 传输在发生传输异常时进行的重试次数。
重试延迟	retry-delay	以毫秒为单位指定每次重试之间的延迟期。 注: Workday 遵守 Retry-After 或 RateLimit-Reset 在外部端点的“429 请求过多”响应中返回一个标头值，前提是该值短于默认最大值 5 分钟，或者短于该值的最大值。retry-delay 您在此处指定的值。 如果缺少响应标头值，Workday 将使用 retry-delay 值 如果两个 retry-delay 和缺少标头值，Workday 将使用 5 分钟的默认值。 如果标头值大于 retry-delay 值和默认最大值。
记录重试次数	log-retries	让您能够将 HTTP 重试尝试记录到服务器日志文件中。

属性A	XML 属性名称	描述
接受 Gzip	accept-gzip	指定是否包括 HTTP accept-encoding=gzip http-out 请求标头中的字段，该字段用于指示是否接受响应中的 gzip 内容。
Gzip 压缩内容	gzip-content	指定是否包括 HTTP content-encoding=gzip http 输出 请求标头中的字段，该字段用于指示是否对传出请求中的消息内容进行 gzip 压缩。
接受	accept	指定 accept 要通过 HTTP 请求发送的标头值，该值定义了 HTTP 服务器响应中可以接受的一个或多个内容类型。

注: 对于 Amazon S3 传输，请为键为“”的 资产管理用户添加标记 *workday-type* 的类型 为，值为 “”集成此标记区分大小写，并且不会忽略空格。

参考信息：FTP-Out 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中 ftp 输出 组件的唯一编号。
将回复发送至	routes-response-to	ftp-out 传输的响应消息的目标。
执行时间	execute-when	一个条件，用于确定是否执行传输。
端点	endpoint	指定要将文件写入到的主机（还可以选择指定端口）和 FTP 目录。 遵循格式 ftp://FTP-directory，其中 FTP-directory 是目标地点。 示例：ftp://ftpserver.example.org/outputDir 注：指定的目录位置必须已存在于 FTP 服务器上。
二进制文件	binary-file	指定是否以二进制格式传输输出文件。
输入文件模式	input-file-pattern	用于匹配特定类型文件的文件名模式。 示例：指定。*.xls 以仅选取 Excel 文件。
输入内容类型	input-content-type	运行时包含的媒体文件类型 content-type 消息标头中的

属性A	XML 属性名称	描述
		值。媒体类型至少包含 2 个部分：类型、子类型以及 1 个或多个可选参数。该 text 子类型有一个可选 charset 参数，用于指定字符编码。 示例：text/html; charset=UTF-8 如果文件是非 XML 文本格式，文件中未明确指定媒体类型和字符集，则此属性特别有用。
输出文件模式	output-file-pattern	指定如何生成输出文件的名称。该模式可以包含文字字符串以及以下一个或多个标记： • \${INFILE} (将替换为输入文件的名称)。 • \${UUID} (将替换为通用唯一编号)。 • \${EXT}，位于模式末尾，后者会根据写入程序集组件输出的数据的内容类型替换为文件扩展名。
方法	method	指定转账方法： • put：将文件写入到包含远程目录的 FTP 端点。 • list：列出 FTP 端点上的所有文件和目录。结合使用“输入文件模式”属性 list 以将模式应用于要返回的列表。您可以访问“”的结果 list 方法 (使用 MVELftp 对象)。示例：ftp.list() 返回一个字符串列表，其中包含最后检索的列表。 • get：会检索第一个与“文件输入模式”属性值匹配的文件。 • delete：会删除第一个与“输入文件模式”属性值匹配的文件。 注：系统不会自动从 FTP 站点中删除文件 get 已执行。如果要删除文件，则必须调用单独的方法。示例：您可以包括一个单独的 ftp-out 专门用于删除文件的传输。
密码	password	与用于访问 FTP 服务器的用户名关联的密码。

属性A	XML 属性名称	描述
用户名	username	用于访问 FTP 服务器的用户名。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
被动模式	passive-mode	指定传输协议是否支持被动模式的 FTP。 在被动模式下，FTP 客户端会发起 FTP 传输所需的两个连接。客户端会通过命令端口连接服务器，并发出 PASV 命令。服务器随即会回复，指明它正在侦听哪个端口来进行数据连接。然后，客户端会发起从其数据端口到指定服务器数据端口的数据连接。最后，服务器会向客户端的数据端口发回 ACK。在主动模式下，客户端会发起命令连接，服务器会发起数据连接。
超时	timeout	以毫秒为单位，指定客户端与传输正在侦听的 FTP 服务器之间进行 FTP 会话的超时期限。 如果您未指定值，会话将在 300000 毫秒（5 分钟）后超时。
多用途互联网邮件扩展 (MIME) 类型映射文件	mime-types-map-file	用于在 MIME 类型和文件扩展名之间进行映射的映射文件的位置。此属性可用于： 当“方法”属性设置为“”时，将 MIME 类型映射到文件扩展名 put。

属性A	XML 属性名称	描述
		如果将“方法”属性设置为“”，则将文件扩展名映射到多用途互联网邮件扩展 (MIME) 类型 get。 用于此映射的文件为 ccx/conf/mime.types。您可以提供此文件的您自己的版本。 此传输可以使用 \${EXT} 令牌置于的末尾 output-file-pattern 值
临时文件名	temp-file-name	指定临时文件名。当“方法”属性设置为“put”，将使用此临时文件名上传文件。FTP 完成后，文件将重命名为“输出文件模式”属性指定的名称。
关闭连接	close-connection	指定 FTP 操作完成时是否发生断开连接。
池大小	pool-size	此组件在每台主机上支持的最大同时连接数和会话数。如果尝试连接到同一主机的线程数量超过指定数量，则系统会强制要求某些线程等待，直至其可以访问池。 要使池大小不受限制，请指定一个负数。
调试	debug	在传输协议中启用 JSMAPE 调试模式

参考信息：FTPS-Out 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中的 ftps-out 组件的唯一编号。
将回复发送至	routes-response-to	ftps-out 传输的响应消息的目标。
执行时间	execute-when	一个条件，用于确定是否执行传输。
端点	endpoint	指定要将文件写入到的主机（还可以选择指定端口）和 FTP 目录。 遵循格式 ftps://FTPS-directory，其中 FTPS-directory 是目标地点。

属性A	XML 属性名称	描述
		示例：ftpserver.example.org/outputDir 注：指定的目录位置在服务器上必须已存在。
二进制文件	binary-file	指定是否将输出文件作为二进制对象进行传输。
显式 SSL	explicit-ssl	指定连接类型：显式 SSL (AUTH TLS) 或隐式 SSL (AUTH SSL)。设置为 true 以使用显式 SLL。 FTP 客户端可以选择像普通 FTP 会话一样继续进行未加密的通信，也可以通过发出 AUTH 命令切换到 FTPS 模式。
输入文件模式	input-file-pattern	用于匹配特定类型文件的文件名模式。 示例：指定。*.xls 以仅选取 Excel 文件。
输入内容类型	input-content-type	运行时包含的媒体文件类型 content-type 消息标头中的值。媒体类型至少包含 2 个部分：类型、子类型以及 1 个或多个可选参数。该 text 子类型有一个可选 charset 参数，用于指定字符编码。 示例：text/html; charset=UTF-8 如果文件是非 XML 文本格式，文件中未明确指定媒体类型和字符集，则此属性特别有用。
输出文件模式	output-file-pattern	指定如何生成输出文件的名称。该模式可以包含文字字符串以及以下一个或多个标记： • \${INFILE} (将替换为输入文件的名称)。 • \${UUID} (将替换为通用唯一编号)。 • \${EXT}，位于模式末尾，后者会根据写入程序集组件输出的数据的内容类型替换为文件扩展名。
方法	method	指定转账方法： • put：将文件写入到包含远程目录的 FTP 端点。 • list：列出 FTP 端点上的所有文件和目录。结合

属性A	XML 属性名称	描述
		使用“输入文件模式”属性 list 以将模式应用于要返回的列表。您可以访问“”的结果 list 方法 (使用 MVELftp 对象)。示例：ftp.list() 返回一个字符串列表，其中包含最后检索的列表。 • get：会检索第一个与“文件输入模式”属性值匹配的文件。 • delete：会删除第一个与“输入文件模式”属性值匹配的文件。 注：系统不会自动从 FTP 站点中删除文件 get 已执行。如果要删除文件，则必须调用单独的方法。示例：您可以包括一个单独的 ftp-out 专门用于删除文件的传输。
密码	password	与用于访问 FTP 服务器的用户名关联的密码。
用户名	username	用于访问 FTP 服务器的用户名。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
被动模式	passive-mode	指定传输协议是否支持被动模式的 FTP。

属性A	XML 属性名称	描述
		在被动模式下，FTP 客户端会发起 FTP 传输所需的两个连接。客户端会通过命令端口连接服务器，并发出 PASV 命令。服务器随即会回复，指明它正在侦听哪个端口来进行数据连接。然后，客户端会发起从其数据端口到指定服务器数据端口的数据连接。最后，服务器会向客户端的数据端口发回 ACK。在主动模式下，客户端会发起命令连接，服务器会发起数据连接。
超时	timeout	以毫秒为单位，指定客户端与传输正在侦听的 FTP 服务器之间进行 FTP 会话的超时期限。 如果您未指定值，会话将在 300000 毫秒（5 分钟）后超时。
多用途互联网邮件扩展 (MIME) 类型映射文件	mime-types-map-file	指定映射文件的位置，该映射文件用于在 MIME 类型和文件扩展名之间进行映射。此属性可用于： - 当“方法”属性设置为“”时，将 MIME 类型映射到文件扩展名 put。 - 如果将“方法”属性设置为“”，则将文件扩展名映射到多用途互联网邮件扩展 (MIME) 类型 get。 用于此映射的文件为 ccx/conf/mime.types。您可以提供此文件的您自己的版本。 此传输可以使用 \${EXT} 令牌置于的末尾 output-file-pattern 值
临时文件名	temp-file-name	指定临时文件名。当“方法”属性设置为“put”，将使用此临时文件名上传文件。FTP 完成后，文件将重命名为“输出文件模式”属性指定的名称。
关闭连接	close-connection	指定 FTP 操作完成时是否发生断开连接。
池大小	pool-size	此组件在每台主机上支持的最大同时连接数和会话数。如果尝试连接到同一主机的线程数量超过指定数量，则系统会强制要求某些线程等待，直至其可以访问池。 要使池大小不受限制，请指定一个负数。

属性A	XML 属性名称	描述
调试	debug	在传输协议中启用 JSMAPE 调试模式

参考信息：SFTP-Out 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中的 sftp-out 组件的唯一编号。
将回复发送至	routes-response-to	sftp-out 传输的响应消息的目标。
执行时间	execute-when	指定一个条件，以确定是否执行传输。
端点	endpoint	指定要将文件写入到的主机（还可以选择指定端口）和 FTP 目录。 遵循格式 sftp://SFTP-directory，其中 SFTP-directory 是目标地点。 示例：sftp://sftpserver.example.org/outputDir 注：指定的目录位置在服务器上必须已存在。
双重身份验证	dual-authentication	指定是否强制使用用户名/密码组合和私钥进行双重身份验证。默认值为 false，这意味着用户可以使用私钥或用户名/密码组合进行身份验证，但不能同时使用两者。 如果该属性设置为 false 并且用户输入私钥（通过配置 sftp-out 组件的私钥属性），则身份验证将自动使用该密钥值和用户名，而忽略密码字段。如果该属性设置为 false 并且未配置私钥，则身份验证将使用用户名/密码组合。 如果该属性设置为 true 则将同时使用私钥值和用户名/密码组合进行身份验证。
主机密钥的指纹	host-key-fingerprint	FTP 服务器将用于 SSH 通信的加密密钥。
用户名	username	用于访问 FTP 服务器的用户名。

属性A	XML 属性名称	描述
密码	password	与用于访问 FTP 服务器的用户名关联的密码。
二进制文件	binary-file	指定是否以二进制格式传输输出文件。
显式 SSL	explicit-ssl	连接类型：显式 SSL (AUTH TLS) 或隐式 SSL (AUTH SSL)。设置为 true 以使用显式 SSL。
输入文件模式	input-file-pattern	用于匹配特定类型文件的文件名模式。 示例：指定。 *.xls 以仅选取 Excel 文件。
输入内容类型	input-content-type	运行时包含的媒体文件类型 content-type 消息标头中的值。媒体类型至少包含 2 个部分：类型、子类型以及 1 个或多个可选参数。该 text 子类型有一个可选 charset 参数，用于指定字符编码。 示例：text/html; charset=UTF-8 如果文件是非 XML 文本格式，文件中未明确指定媒体类型和字符集，则此属性特别有用。
输出文件模式	output-file-pattern	指定如何生成输出文件的名称。该模式可以包含文字字符串以及以下一个或多个标记： • \${INFILE} (将替换为输入文件的名称)。 • \${UUID} (将替换为通用唯一编号)。 • \${EXT}，位于模式末尾，后者会根据写入程序集组件输出的数据的内容类型替换为文件扩展名。
方法	method	指定转账方法： • put：将文件写入到包含远程目录的 FTP 端点。 • list：列出 FTP 端点上的所有文件。结合使用“输入文件模式”属性 list 以将模式应用于要返回的列表。您可以访问“”的结果 list 方法 (使用 MVELftp 对象)。示例：ftp.list() 返回一个字符串列表，其中包含最后检索的列表。

属性A	XML 属性名称	描述
		• <code>get</code> : 会检索第一个与“文件输入模式”属性值匹配的文件。 • <code>delete</code> : 会删除第一个与“输入文件模式”属性值匹配的文件。 注: 系统不会自动从 FTP 站点中删除文件 <code>get</code> 已执行。如果要删除文件, 则必须调用单独的方法。示例: 您可以包括一个单独的 <code>ftp-out</code> 专门用于删除文件的传输。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	<code>store-message</code>	除非 Workday 技术支持另行通知, 否则 无 指定内容。 要存储消息, 请使用 Workday Studio 的“store”步骤。
传输类	<code>transport-class</code>	如果需要, 为每个装配件指定一个备选的实现类。 您可以通过编辑 <code>WEB-INF/classes/spring/assembly-bean.xml</code> 文件来全局更改实现类。
忽略动态端点	<code>ignore-dynamic-endpoints</code>	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值, 还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
超时	<code>timeout</code>	以毫秒为单位, 指定客户端与传输正在侦听的 FTP 服务器之间进行 FTP 会话的超时期限。 如果您未指定值, 会话将在 300000 毫秒 (5 分钟) 后超时。
多用途互联网邮件扩展 (MIME) 类型映射文件	<code>mime-types-map-file</code>	指定映射文件的位置, 该映射文件用于在 MIME 类型和文件扩展名之间进行映射。此属性可用于: • 当“方法”属性设置为“”时, 将 MIME 类型映射到文件扩展名 <code>put</code>。 • 如果将“方法”属性设置为“”, 则将文件扩展名映

属性A	XML 属性名称	描述
		射到多用途互联网邮件扩展 (MIME) 类型 get。 用于此映射的文件为 ccx/conf/mime.types。您可以提供此文件的您自己的版本。 此传输可以使用 \${EXT} 令牌置于 的末尾 output-file-pattern 值
临时文件名	temp-file-name	指定临时文件名。当“方法”属性设置为“put”，将使用此临时文件名上传文件。FTP 完成后，文件将重命名为“输出文件模式”属性指定的名称。
关闭连接	close-connection	指定 FTP 操作完成时是否发生断开连接。
池大小	pool-size	此组件在每台主机上支持的最大同时连接数和会话数。如果尝试连接到同一主机的线程数量超过指定数量，则系统会强制要求某些线程等待，直至其可以访问池。 要使池大小不受限制，请指定一个负数。
调试	debug	在传输协议中启用 JSMAPE 调试模式

参考信息：XMPP-Out 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中 xmpp-out 组件的唯一编号。
将回复发送至	routes-response-to	ftp-out 传输的响应消息的目标。
执行时间	execute-when	一个条件，用于确定是否执行传输。
端点	endpoint	指定即时消息的接收人。示例： xmpp:logan.mcneil@gmail.com
域	domain	注册用户名的域。示例： gmail.com
服务器	server	XMPP/Jobs 服务器的主机或 IP 地址。
端口	port	XMPP 传输侦听消息的端口号。大多数 XMPP 服务器使用端口 5222。

属性A	XML 属性名称	描述
用户名	username	用于访问 FTP 服务器的用户名。
密码	password	与用于访问 FTP 服务器的用户名关联的密码。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。

参考信息：Custom-Out 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中 自定义输出 构成的唯一编号。
将回复发送至	routes-response-to	使用 自定义输出 传输方式的响应消息的目标。
执行时间	execute-when	一个条件，用于确定是否执行传输。
端点	endpoint	自定义输出 传输协议将消息发送到的端点。由“custom-out”Spring Bean 实现来使用。
输入	input	指定装件组件从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消

属性A	XML 属性名称	描述
		息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i> : 消息中的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i> : MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。 • <i>变量</i> : 使用中的已命名变量 <i>MediationContext</i>。
输出	output	指定组件将输出定向到何处。共有 5 个可能的值： • <i>message</i> : 整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 • <i>soapbody</i> : 中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i> : 消息的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i> : MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 • <i>变量</i> : 将输出另存为中的已命名变量 <i>MediationContext</i>。
java - java - java	spring-bean	指定用于实现传输接口类的 <i>javax</i> 。
方法名称	method-name	自定义 Java 类中 Java 方法的名称。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-

属性A	XML 属性名称	描述
		bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。

参考信息：Email-Out 传输属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中通过 电子邮件发出的组件的唯一编号。
将回复发送至	routes-response-to	通过 电子邮件发出的 传输协议的响应消息目标。
执行时间	execute-when	一个条件，用于确定是否执行传输。
如要执行以下操作	to	意向收件人的常用电子邮件地址。遵循格式 mailto:[emailaddress]。 如果只想使用密件抄送地址，请指定一个空地址。mailto: 值
主题	subject	电子邮件的主题行。
自	from	在发送的电子邮件的“发件人”字段中应显示的电子邮件地址。
Ssl	ssl	指定连接是否采用 SSL。
Starttls	starttls	指定是否通过在启动 SMTP 会话前发出 STARTTLS 命令来激活与远程 SMTP 服务器的即时 SSL 协商。SMTP 服务器将使用以下其中一种应答代码进行响应： • 220 Ready to start • TLS 501 Syntax error (no parameters allowed) • 454 TLS not available due to a temporary reason 注：“Ssl”属性和“Starttls”属性是互斥的。
主机	host	传输层用于传送邮件的 SMTP 邮件服务器的名称或 IP 地址。

属性A	XML 属性名称	描述
		如果您使用 OAuth，请将此值设置为 localhost。
端口	port	运行邮件服务器的端口号。默认值为 25，即标准 SMTP 端口。 异常： - 如果 ssl 属性设置为 true 并且未指定端口，则默认端口为 465。 - 如果 StartTLS 属性设置为 true 并且未指定端口，则默认端口为 587。 注:
用户		用于访问电子邮件服务器的用户名。
密码	password	在 User 属性中指定的用户名的密码。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
Bcc	bcc	使用英文分号分隔符指定密送电子邮件收件人。示例： logan.mcneil@workday.com; jake.lee@workday.com。
Cc	cc	指定抄送电子邮件收件人，使用英文分号分隔符。示例： logan.mcneil@workday.com; jake.lee@workday.com。

属性A	XML 属性名称	描述
回复到	reply-to	此电子邮件地址会在用户回复电子邮件消息时自动插入“收件人”字段。对于大多数电子邮件，只需使用“From”字段中的地址即可回复，因此无需定义“Reply To”属性。指定多个电子邮件地址时，请使用英文分号分隔符。示例： logan.mcneil@workday.com; jake.lee@workday.com。
超时	timeout	为客户端与传输正在侦听的 FTP 服务器之间的 FTP 会话指定超时期限（以毫秒为单位）。如果您未指定值，会话将在 300000 毫秒（5 分钟）后超时。
暂时性异常	transient-exceptions	暂时性异常是指在重试时无需任何修改即可成功的异常。非暂时性异常是指在从问题的根本原因得到解决之前，在重试时继续失败的异常。 默认情况下，Workday 仅处理 <code>java.net.ConnectException</code> 与 <code>java.net.SocketException</code> 将错误视为暂时性异常。 以逗号、空格或以逗号和空格分隔的类名称枚举形式指定其他暂时性异常。示例： <code>java.net.UnknownHostException</code> <code>java.net.ConnectException</code> <code>java.net.SocketException</code> <code>java.net.UnknownHostException,j</code> <code>java.net.UnknownHostException,</code> <code>java.net.ConnectException,</code> <code>java.net.SocketException</code> <code>@{new</code> <code>java.net.UnknownHostException(</code> 注：如果您指定了其他暂时性异常，则还必须包括默认异常。 <code>java.net.ConnectException</code> 与 <code>java.net.SocketException</code> “”列表中的“”。否则，Workday 将覆盖它们。

“SMTP Headers”选项卡

属性	XML 属性名称	描述
名称	name	自定义 SMTP 标头的名称。

属性	XML 属性名称	描述
		要在任何现有行下方添加新的自定义标头，请单击“Add”，然后提供“Name”和“Value”。 要在现有行上方添加新标头，请将光标放在该行中，然后单击“Add”下拉菜单中的“Insert”。 使用“SMTP Headers”选项卡右侧的按钮可删除行或对行重新排序。
值	value	自定义 SMTP 标头的值。

OAuth 选项卡

要将 Microsoft Outlook OAuth 身份验证与 Studio 的电子邮件发送组件结合使用，请将 Studio 作为应用注册到 Microsoft Azure，提供域范围内的委派，然后向管理员授予许可 Mail.Send 权限和获取各种值。有关详细信息，请参见 <https://learn.microsoft.com/en-gb/azure/active-directory/develop/v2-oauth2-client-creds-grant-flow>。

要将 Google Gmail OAuth 身份验证与 Workday Studio 的 Email-Out 构成结合使用，请在 Google Cloud 中注册服务账户，提供域范围的委派，并获取各种值的 JSON 描述。有关详细信息，请参见 <https://support.google.com/cloud/answer/6158849?hl=en>。

属性	XML 属性名称	描述				
提供商	provider	电子邮件提供商的名称（以下任一值）MICROSOFT_OUTLOOK 或 GOOGLE_GMAIL。				
名称	name	Workday 规定的名称，该名称会将您与从 Microsoft Azure 门户或 Google Cloud Console 获取的各种值相关联。 对于 Microsoft Outlook，请输入以下名称： - outlook.client.id - Outlook.client.secret - outlook.tenant.id - outlook.user.id 对于 Google Gmail，请输入： - gmail.service.account.key.js - gmail.delegated.user				
值	value	从 Microsoft 或 Google 获取且与上述名称相关联的值。 对于 Microsoft Outlook，请创建以下名称/值对： <table><tr><th>名称</th><th>值</th></tr><tr><td>outlook.client.id</td><td>在 Microsoft Azure 门户上已注册应用</td></tr></table>	名称	值	outlook.client.id	在 Microsoft Azure 门户上已注册应用
名称	值					
outlook.client.id	在 Microsoft Azure 门户上已注册应用					

属性	XML 属性名称	描述	
		名称	值
			的“Overview”页面中，创建： 应用程序（客户端）编号。
		Outlook.client.secret	在 Microsoft Azure 门户上已注册应用的“Certificates & secrets”页面中，创建： 客户端密码编号。
		outlook.tenant.id	在 Microsoft Azure 门户上已注册应用的“Overview”页面中，创建： 目录（租户）编号。
		outlook.user.id	用于发送电子邮件的 Outlook 账户地址。同时也会在“Common”选项卡的“From”字段中输入。
		对于 Google Gmail，请创建以下名称/值对：	
		名称	值
		gmail.service.account.json	从 Google Cloud 控制台上“Service accounts”页面的“KEYS”版块下载的文件中的 JSON 文本。
		gmail.delegated.user	用于发送电子邮件

属性	XML 属性名称	描述	
		名称	值
			的 Gmail 地址。必须是与 Google 服务账户关联的主体。同时也会在“Common”选项卡的“From”字段中输入。

Gmail over SSL 或 Gmail with STARTTLS

在将“email-out”传输配置为使用 Gmail over SSL 或 Gmail with STARTTLS 时，请确保：

- 用户名 和 密码 包含有效的Gmail 凭据。
- 端点 是 `mailto:[name]@gmail.com`。
- 服务提供者 是 `smtp.gmail.com`。
- `ssl` 或 `startTLS` 设置为 `true`。
- 端口 为以下任意一种端口：
 - 465，如果 `ssl` 设置为 `true`。
 - 587，如果 `startTLS` 设置为 `true`。
- 超时 为 30000（30 秒）。

Microsoft Office 365 over SSL 或 STARTTLS

在将“email-out”传输配置为使用 Microsoft Office 365 over SSL 或 Microsoft Office 365 with STARTTLS 时，请确保：

- 用户名 和 密码 包含有效的 Office 365 用户凭据。
- 端点 是 `mailto:[name]@[domain-name.top-level-domain]`。
- 服务提供者 是 `smtp.office365.com`。
- `ssl` 或 `startTLS` 设置为 `true`。
- 端口：
 - 465，如果 `ssl` 设置为 `true`。
 - 587（如果将 `starttls` 设置为 `true`）。
- 超时 为 30000（30 秒）。

配置非二进制格式的消息正文和附件

在接收传入消息时，许多电子邮件应用程序会匹配多用途互联网邮件扩展 (MIME) 消息中的第一个文本部分，并将其用作消息文本。若要将根消息部件的 MIME 类型设置为 `text/plain`，将消息的 输入 和 输出 属性设置为 消息 ()，然后将“输出的多用途 互联网邮件扩展 (MIME 类型) 类型”设置为“文本/明文”。

通过 消息构建器，使用 文本 元素为电子邮件的消息正文定义消息内容。

对于附件，将“输出”属性“”设置为“”，并将“输出”附件索引设置为“”。0，然后将“输出 MIME 类型”设置为 `text/plain`。

同样，通过 消息构建器，使用 文本 元素来定义附件的消息内容。

为防止某些电子邮件应用程序在消息正文中显示文本附件，即使附件实际上是文本，也请将附件的 MIME 类型设置为“`application/octet-stream`”。为此，请在 评估 步骤中使用 MVEL 表达式。示例：
`parts[1].setMimeType('application/octet-stream')`

如要设置附件的文件名，请使用 MVEL 表达式来设置 Content-Disposition 标头。示例：
`parts[1].setHeader('Content-Disposition', 'attachment; filename=payload.txt')`。
最后，配置通过 电子邮件发送的 步骤。

为二进制格式配置消息正文和附件
要将二进制文件注入“MediationContext”，请为从 FTP 或 SFTP 传输中检索文件的“workday-in”定义一项检索服务。

创建消息正文，然后使用“eval”步骤将检索到的文件复制到变量中。您可以使用“da”文档访问器变量和 toVar 方法来实现此目的。示例：`da.toVar(Rising_2011_Logo.jpg', 'wd.retrieve.variable')`。

使用 复制 步骤从以下位置复制文件：`wd.retrieve.variable` 添加到附件”为此，请将 输入 属性设置为变量，并指定 `wd.retrieve.variable` 作为输入变量。将“输出”属性设置为“附件”，指定的 输出 附件索引为 0，“输出多用途 互联网邮件扩展 (MIME) 类型”为“图片/jpeg”。

最后，配置通过 电子邮件发送的 步骤。

HTTP-Out 传输子元素

概念：HTTP-Out 传输子元素

“http-out”传输支持以下子元素：

子元素	描述
http-basic-auth	支持对“http-out”传输进行基本身份验证。
http-custom-auth	支持对“http-out”传输进行自定义身份验证。
https-properties	支持“http-out”传输中的 HTTPS 属性。

参考信息：HTTP-Basic-Auth 属性

“Common”选项卡

属性A	XML 属性名称	描述
密码	password	用于身份验证的密码。
用户名	username	用于身份验证的用户名。
领域	realm	回调处理程序 bean 使用的领域名称。
回调处理程序 bean	callback-handler-bean	一个实现 <code>javax.security.auth.callback.Callback</code> 并处理回调类型 <code>com.capeclear.transport.impl.http.a</code> 可提供用户名和密码详细信息以进行身份验证。 如果您不为“http-Basic-auth”子元素指定密码详细信息，则需要改用回调处理程序 bean。示例：<code><http-basic-auth username="me" callback-handler-</code>

属性A	XML 属性名称	描述
		bean="myCallBackHandlerBean"/>。 如果您没有为 “http-Basic-auth” 子元素指定用户名和密码详细信息，则需要改用回调处理程序 bean。示例：<http-basic-auth callback-handler-bean="myCallBackHandlerBean"/>。 如果您为 “http- Basic-auth” 子元素指定用户名和密码详细信息，则不需要回调处理程序 bean。示例：<http-basic-auth username="me" password="mypassword"/>。

参考信息：HTTP-Custom-Auth 属性

“Common”选项卡

属性A	XML 属性名称	描述
密码	password	用于身份验证的密码。
用户名	username	用于身份验证的用户名。
领域	realm	回调处理程序 bean 使用的领域名称。
回调处理程序 bean	callback-handler-bean	一个实现 javax.security.auth.callback.Callback 并处理回调类型 com.capeclear.transport.impl.http.a 可提供用户名和密码详细信息以进行身份验证。 如果您不为 “http-custom-auth” 子元素指定密码详细信息，则需要改用回调处理程序 bean。示例：<http-custom-auth username="me" callback-handler-bean="myCallBackHandlerBean"/>。 如果您没有为 “http-custom-auth” 子元素指定用户名和密码详细信息，则需要改用回调处理程序 bean。示例：<http-custom-auth callback-handler-bean="myCallBackHandlerBean"/>。 如果您为 “http-custom-auth” 子元素指定用户名和密码详细

属性A	XML 属性名称	描述
		信息，则不需要回调处理程序 bean。示例：<http-custom-auth username="me" password="mypassword"/>。
Authenticator Cube	authenticator-bean	执行自定义身份验证的春季 bean。

参考信息：HTTPS 属性

“Common”选项卡

属性A	XML 属性名称	描述
接受服务提供者	accept-hosts	要用于验证 SSL 证书中指定的主机名的 Java 正则表达式字符串。
密钥密码	key-password	访问密钥库中相应密钥时要使用的密码。
密钥库文件	key-file	连接到 HTTPS 端点时使用的密钥库文件的位置。
密钥库密码	keystore-password	访问密钥库时使用的密码。
代理主机	proxy-host	代理服务器的主机名，该服务器用于代理此传输的 HTTPS 请求。
代理端口	proxy-port	代理服务器的端口，您使用该代理服务器来代理此传输的 HTTPS 请求。
信托库文件	truststore-file	连接到 HTTPS 端点时使用的信任库文件的位置。
信托库密码	truststore-password	访问信任库时要使用的密码。

FTPS-Out 传输子元素

概念：FTPS-Out 传输子元素

“ftps-out”传输支持以下子元素：

子元素	描述	注释
proxy-properties	代理身份验证属性。	
firewall-properties	防火墙连接属性。	默认情况下，“ftps-out”传输与防火墙相容，因为其所用连接属于被动连接。但某些防火墙可能会要求您通过它们路由所有连接。

子元素	描述	注释
client-key-properties	客户端密钥身份验证属性。	
server-key-properties	服务器密钥身份验证属性。	

参考信息：Proxy-Properties 属性

“Common”选项卡

属性A	XML 属性名称	描述
代理主机	proxy-host	连接到安全 FTP 站点时要使用的代理服务器的主机名。
代理密码	proxy-password	代理服务器在连接到安全 FTP 站点时使用的密码。
代理端口	proxy-port	连接到安全 FTP 站点时要使用的代理服务器端口号。
代理类型	proxy-type	连接到安全 FTP 站点时要使用的代理服务器的代理类型。默认类型为 http。
代理用户名	proxy-username	连接到安全 FTP 站点时要使用的代理服务器用户名。

参考信息：Firewall-Properties 属性

“Common”选项卡

属性A	XML 属性名称	描述
防火墙主机	firewall-host	连接到安全 FTP 站点时要使用的防火墙的主机名或 IP 地址。
防火墙端口	firewall-port	防火墙的端口号。

参考信息：Client-Key 属性

“Common”选项卡

属性A	XML 属性名称	描述
密码	passphrase	对安全 FTP 站点进行身份验证时使用的私钥密码。
私钥	private-key	与用户名关联的私钥，用于 SFTP 身份验证。您可通过以下 2 种方式指定该属性： 使用附件协议和 Workday 编号，以 URI 的形式提供。示例： attachment:privatekey/@{WID}，其中 WID 是 32 个字符的 Workday 编号。

属性A	XML 属性名称	描述
		以相对路径位置表示。示例：<code>private_key.txt</code>，其中 <code>private_key.txt</code> 是位于 <code>ws/WEAR-INF</code> 文件夹中的文件。

参考信息：Server-Key 属性

“Common”选项卡

属性A	XML 属性名称	描述
密码	passphrase	对安全 FTP 站点进行身份验证时使用的私钥密码。
私钥	private-key	与用户名关联的私钥，用于 SFTP 身份验证。您可通过以下 2 种方式指定该属性： - 使用附件协议和 Workday 编号，以 URI 的形式提供。示例：<code>attachment:privatekey/{WID}</code>，其中 <code>WID</code> 是 32 个字符的 Workday 编号。 - 以相对路径位置表示。示例：<code>private_key.txt</code>，其中 <code>private_key.txt</code> 是位于 <code>ws/WEAR-INF</code> 文件夹中的文件。

SFTP-Out 子元素

概念：SFTP-Out 子元素

“sftp-out”传输支持以下子元素：

子元素	描述	注释
proxy-properties	代理身份验证属性。	
private-key-properties	私钥身份验证属性。	

参考信息：Proxy-Properties 属性

“Common”选项卡

属性A	XML 属性名称	描述
代理主机	proxy-host	连接到安全 FTP 站点时要使用的代理服务器的主机名。

属性A	XML 属性名称	描述
代理密码	proxy-password	代理服务器在连接到安全 FTP 站点时使用的密码。
代理端口	proxy-port	连接到安全 FTP 站点时要使用的代理服务器端口号。
代理类型	proxy-type	连接到安全 FTP 站点时要使用的代理服务器的代理类型。默认类型为 http。
代理用户名	proxy-username	连接到安全 FTP 站点时要使用的代理服务器用户名。

参考信息：Private-Key 属性

“Common”选项卡

属性A	XML 属性名称	描述
密码	passphrase	对安全 FTP 站点进行身份验证时使用的私钥密码。
私钥	private-key	与用户名关联的私钥，用于 SFTP 身份验证。您可以将其指定为相对路径位置。示例：private_key.txt，其中私钥文件位于 ws/WSAR-INF 文件夹。

构成

概念：构成

构成使您能够在消息通过您的集成时对其进行操作和转换。Workday Studio 为程序集“Palette”的“Components”版块中的以下构成提供支持：

中介构成	描述	注释
async-mediation	支持 1 个处理链，您可以使用步骤填充该处理链，以在程序集中定义请求消息。	
sync-mediation	支持 2 个处理链，您可以使用步骤填充这些处理链，以在程序集中定义请求消息和请求-响应消息。	
custom-mediation	创建一个同时支持请求消息和请求-响应消息的可自定义构成。	
route	实现路由策略和子路由，以引导消息在您的程序集中流转。	
splitter	实现拆分器策略和子路由，以引导消息在您的程序集中流转。	向下游“aggregator”构成发送信号。

中介构成	描述	注释
aggregator	将消息串联成批次，您可以将这些批次定向至已配置的目的地。	检测上游“splitter”构成。

参考信息：Async-Mediation 构成属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中的 异步中介 组件的唯一编号。
寄送目的地	routes-to	async-mediation 将消息定向到的集合元素。
执行时间	execute-when	此条件用于确定是否执行 异步中介 组件。

“Advanced”选项卡

属性A	XML 属性名称	描述
出错后继续	continue-after-error	指示在错误处理程序清除错误时 Workday 的行为。 默认值为“rewind”，这意味着从程序集中的上一个构成继续进行处理。 如果设置为“recover”，则 Workday 会在发生错误时跳至中介内的下一个步骤。
处理下游错误	handle-downstream-errors	指示是否处理下游错误。 如果设置为“false”，则本地错误处理程序仅处理在“async-mediation”构成内发生的错误。而全局错误处理程序将会处理下游错误。 如果设置为“true”，则“async-mediation”构成的本地错误处理程序可以处理下游错误。

参考信息：Sync-Mediation 构成属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中 同步中介 组件的唯一编号。
寄送目的地	routes-to	sync-Mediation 将消息定向到的程序集元素。

属性A	XML 属性名称	描述
将回复发送至	routes-response-to	sync-Mediation 组件的响应消息的目标。
执行时间	execute-when	此条件用于确定是否执行 同步中介 组件。

“Advanced”选项卡

属性A	XML 属性名称	描述
出错后继续	continue-after-error	指示在错误处理程序清除错误时 Workday 的行为。 默认值为“rewind”，这意味着从程序集中的上一个构成继续进行处理。 如果设置为“recover”，则 Workday 会在发生错误时跳至中介内的下一个步骤。
处理下游错误	handle-downstream-errors	指示是否处理下游错误。 如果设置为“false”，则本地错误处理程序仅处理在“sync-mediation”构成内发生的错误。而全局错误处理程序将会处理下游错误。 如果设置为“true”，则“sync-mediation”构成的本地错误处理程序可以处理下游错误。

参考信息：Custom-Mediation 构成属性

“Common”选项卡

属性A	XML 属性名称	描述
Id		“custom-mediation”构成在程序集中的唯一编号。
寄送目的地	routes-to	自定义中介 将消息定向到的组合件元素。
将回复发送至	routes-response-to	自定义中介 组件的响应消息的目标。
执行时间	execute-when	此条件用于确定是否执行 自定义中介 组件。
java - java - java	ref	由 custom-mediation实施的自定义 Spingle Java 类。
申请方式	request-method	用于请求处理的方法的名称。
回复方法	response-method	用于响应处理的方法的名称。

参考信息：Route 构成属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	装配件内的 路线 构成唯一编号。

参考信息：Splitter 构成属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	拆分器 组件在程序集中的唯一编号。
无拆分消息错误	no-split-error-message	指定当 拆分器 组件没有为批量输入的消息生成拆分消息时，Workday 是否引发错误。
拆分截止日期	split-until	指定拆分器组件何时停止处理批量输入的消息。

参考信息：Aggregator 构成属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集内 聚合器 构成的唯一编号。
寄送目的地	routes-to	聚合者 将消息定向到的集合元素。
核对时间	collate-when	一个 MVEL 表达式，用于确定是否对传入的消息进行核对。
根据最后一条消息强制批处理	force-batch-on-last-message	拆分器 组件收到最后一条消息后强制进行批处理。
强制批处理的条件	force-batch-when	一个 MVEL 表达式，用于确定是否对传入消息进行批处理。

“Advanced”选项卡

属性A	XML 属性名称	描述
允许外部控制	allow-external-control	使上游组件能够控制 聚合器 组件何时进行核对和/或批处理。

Route 构成子元素

概念：Route 构成子元素

“route”构成支持以下子元素。

子元素	描述	注释
custom-strategy	您配置为 Spring Bean 的可自定义的路由策略。	
regex-strategy	实现将消息定向至子路由的正则表达式。	实现“choose-route”子元素，以指明您的目标“sub-route”。
mvel-strategy	实现将消息定向至子路由的 MVEL 表达式。	实现“choose-route”子元素，以指明您的目标“sub-route”。
round-robin-strategy	按周期性顺序将消息定向至子路由。	
xpath-strategy	实现将消息定向至子路由的 XPath 表达式。	实现“choose-route”子元素，以指明您的目标“sub-route”。
failover-strategy	实现将消息定向至子路由的故障转移机制。如果某个“sub-route”子元素失败，则下一个“sub-route”将替换该子元素。	
all-strategy	将消息依次定向至每个子路由。	
doc-iterator	遍历您的程序集从 Workday 检索的文档。	
loop-strategy	实现将消息定向至子路由的循环机制。	
choose-route	指示您的策略将消息定向至的目标“sub-route”子元素。	支持 XPath、MVEL 或正则表达式。
sub-route	接收来自路由策略的消息，并将这些消息定向至程序集中的其他构成和传输。	

参考信息：Custom-Strategy 属性

“Common”选项卡

属性A	XML 属性名称	描述
java - java - java	ref	用于实施您的 custom-Strategy 接口类的 Spingle。
可重复	repeatable	启用循环播放。 如果设置为“true”，程序集将实现一个循环，以重复调用您的“custom-strategy”。

属性A	XML 属性名称	描述
		如果设置为“false”，则程序集不会为您的“custom-strategy”实现循环。
重复限制	repeat-limit	限制循环调用您的 custom-strategy 的次数。
处理错误	handle-error	指明父级 路由 “”的构成是否处理子路由“”的异常。

参考信息：Regex-Strategy 属性

“Common”选项卡

属性A	XML 属性名称	描述
“筛选器”	filter	指明当您在“choice-route”子元素中实施无效的正则表达式时，Workday 是否显示错误消息。

参考信息：MVEL-Strategy 属性

“Common”选项卡

属性A	XML 属性名称	描述
“筛选器”	filter	指明当您在 直属 子元素中实施无效的 MVEL 表达式时，Studio 是否会显示错误消息。

参考信息：Round-Robin-Strategy 属性

“Common”选项卡

属性A	XML 属性名称	描述
复制申请	clone-request	指示 round-robin-strategy 是否向每个 子路由 子级元素发送原始中介消息的副本。
故障转移	failover	指示 round-robin-strategy 在遇到失败的 子路由时是否会循环访问您的 子级路由 “”。
超时	timeout	一个毫秒值，表示从流转中排除失败的子流转多长时间。

参考信息：XPath-Strategy 属性

“Common”选项卡

属性A	XML 属性名称	描述
“筛选器”	filter	指明当您在 直属 子元素中实施无效的 XPath 表达式时，Workday 是否显示错误消息。
命名空间	namespaces	以下格式中 XPath 表达式使用的命名空间 prefix namespace prefix2 namespace2。重设Assassing.xml 文件中的命名空间映射。
XPath 版本	xpath-version	指定使用 XPath 1.0、2.0 还是 3.0 版本。对于 Workday 11 及之前的程序集版本，请输入值 1。对于 Workday 12 及之后的程序集版本，请输入值 2 或 3。 您可以允许 Workday 使用可插入系统属性值的 MVEL 模板动态设置版本。示例： @{props.get('the_version')}。

参考信息：Failover-Strategy 属性

“Common”选项卡

属性A	XML 属性名称	描述
清除故障	clear-fault	指示 failover-strategy 是否会清除由子 路由子级 元素生成的异常。
复制申请	clone-request	指示 failover-strategy 是否向每个子 路由 子元素发送原始中介消息的副本。

参考信息：All-Strategy 属性

“Common”选项卡

属性A	XML 属性名称	描述
复制申请	clone-request	指明 all-策略 是否向每个子 传递 子元素发送原始中介消息的副本。
忽略错误	ignore-errors	指明 all-策略 是否忽略子 路由子级 元素生成的错误。

参考信息：Doc-Iterator 属性

“Common”选项卡

属性A	XML 属性名称	描述
变量名称	variable-name	文档迭代器 用于存储检索到的文档的变量名称。默认值为 wd.retrieve.variable。
标签	labels	指定附加到您的检索文档的一个或多个标签。
排序依据	sort-by	用于检索和呈现文档的排序顺序。选项包括： - FILENAME_ASCENDING - FILENAME_DSCENDING - NONE

参考信息：循环策略属性

“Common”选项卡

属性A	XML 属性名称	描述
条件	condition	生成布尔值的 MVEL 表达式。“loop-strategy”将一直运行，直到表达式的计算结果为假。
增量	increment	每次循环迭代的增量值。
初始化	init	循环的起始值。
重复限制	repeat-limit	由 循环策略实施的最大循环数。 一个路径策略总共可以重复最多 200,000 次。如果多次调用同一策略，系统不会重置循环计数器，而是从上一个值继续。您可以使用自定义 Java Bean 或自定义拆分器来避免此限制。

参考信息：Choose-Route 属性

“Common”选项卡

属性A	XML 属性名称	描述
表达式	expression	用于匹配传入消息值的表达式。
路由	route	匹配表达式值的目标子 路由 子元素。

参考信息：Sub-Route 属性

“Common”选项卡

属性A	XML 属性名称	描述
名称	name	装配件中的 子路线 的唯一名称。
寄送目的地	routes-to	消息 子路线的 目标集合元素。

Splitter 构成子元素

概念：Splitter 构成子元素

“splitter”构成支持以下子元素。

子元素	描述	注释
custom-splitter	您配置为 Spring Bean 的可自定义的拆分器策略。	
standard-splitter	从批量消息中提取数据，将这些数据拆分为多个令牌，然后将令牌化的数据引导至您的目标子路由。	支持以下子元素： - header-ends-with - header-fixed-lines - content-fixed-lines - content-starts-ends - content-starts-with - footer-fixed-lines - footer-starts-with
xpath-splitter	实现 XPath 表达式，以从传入的批量消息中提取数据。	
xml-stream-splitter	实现 XPath 表达式，以从传入的批量消息中提取数据。支持 XML 流式传输。	
json-splitter	从批量输入消息中提取 JSON。	
unzip-splitter	从 ZIP 和 TAR 文件中提取数据，将这些数据拆分为单独的消息，然后将消息定向到其目标子路由。	
mtable-splitter	从“”中提取数据 mtable 对象，并将数据拆分为单独的行。	
子路线	接收来自拆分器策略的消息，并将这些消息定向到程序集中的其他构成和传输。	

参考信息：Custom-Splitter 属性

“Common”选项卡

属性A	XML 属性名称	描述
java - java - java	ref	用于实现 自定义拆分器 接口类的 Spingle。

参考信息：Standard-Splitter 属性

“Common”选项卡

属性A	XML 属性名称	描述
输出 MIME 类型	output-mimetype	令牌化输出数据的 MIME 类型。
行令牌分隔符正则表达式	line-token-delimiter-regex	分隔正则表达式，用于从批量消息中提取令牌化数据。
多行分隔符	multiple-lines-separator	一个字符串，用于连接跨多行的标记。

参考信息：XPath-Splitter 属性

“Common”选项卡

属性A	XML 属性名称	描述
命名空间	namespaces	XPath 表达式使用的命名空间。
XPath	xpath	用于确定拆分输入消息的位置的 XPath 表达式。
XPath 版本	xpath-version	指定使用 XPath 1.0、2.0 还是 3.0 版本。对于 Workday 11 及之前的程序集版本，请输入值 1。对于 Workday 12 及之后的程序集版本，请输入值 2 或 3。 您可以允许 Workday 使用可插入属性值的 MVEL 模板动态设置版本。示例： @{props.get('the_version')}。

参考信息：XML-Stream-Splitter 属性

“Common”选项卡

属性A	XML 属性名称	描述
命名空间	namespaces	XPath 表达式使用的命名空间。
XPath	xpath	用于从批量输入消息中提取 XML 的 XPath 表达式。
XPath 版本	xpath-version	指定使用 XPath 1.0、2.0 还是 3.0 版本。对于 Workday 11 及

属性A	XML 属性名称	描述
		之前的程序集版本，请输入值 1。对于 Workday 12 及之后的程序集版本，请输入值 2 或 3。 您可以允许 Workday 使用可插入系统属性值的 MVEL 模板动态设置版本。示例： @{props.get('the_version')}}。

参考信息：JSON-Splitter 属性

“Common”选项卡

属性A	XML 属性名称	描述
Json 路径	json-path	用于从批量输入消息中提取 JSON 的表达式。

参考信息：Unzip-Splitter 属性

“Common”选项卡

属性A	XML 属性名称	描述
文件模式	file-pattern	从 ZIP 和 TAR 输入文件中提取数据的正则表达式。
格式	format	输入文件的格式。有以下选项： - zip - tar

参考信息：Mtable-Splitter 属性

“Common”选项卡

属性A	XML 属性名称	描述
表名称	mtable-name	的名称 MediationContext 属性，其中包含 mtable。
行名称	row-name	的名称 MediationContext 属性（存储被迭代的行）。
行格式	row-format	确定已迭代行的类。支持的值包括： - array - map

Standard-Splitter 子元素

概念：Standard-Splitter 子元素

“standard-splitter”构成支持以下子元素。

子元素	描述
header-ends-with	指示文档中标头板块的结尾。
header-fixed-lines	指示文档中标头板块所占的行数。
content-fixed-lines	指示文档中数据记录内容板块所占的行数。
content-starts-ends	指示文档中数据记录内容板块的起点和终点。
content-starts-with	指示文档中数据记录内容板块的起点。
footer-fixed-lines	指示文档中尾部板块所占的行数。
footer-starts-with	指示文档中尾部板块的起点。

参考信息：Header-Ends-With 属性

“Common”选项卡

属性A	XML 属性名称	描述
包括在标头中	include-in-header	确定正则表达式的匹配行是否是标头的一部分。
正则表达式	regexp	标识标头结尾的正则表达式。

参考信息：Header-Fixed-Lines 属性

“Common”选项卡

属性A	XML 属性名称	描述
行计数	lines-count	标头占用的行数。

参考信息：Content-Fixed-Lines 属性

“Common”选项卡

属性A	XML 属性名称	描述
忽略行 - 正则表达式	ignore-lines-regexp	一个正则表达式，用于从数据记录内容板块中排除匹配的行。
行计数	lines-count	数据记录内容板块占用的行数。

参考信息：Content-Starts-Ends 属性

“Common”选项卡

属性A	XML 属性名称	描述
结束正则表达式	end-regexp	一个正则表达式，用于标识数据记录内容板块的末尾。
启动正则表达式	start-regexp	一个正则表达式，用于标识数据记录内容板块的开头。

参考信息：Content-Starts-With 属性

“Common”选项卡

属性A	XML 属性名称	描述
正则表达式	regexp	一个正则表达式，用于标识数据记录内容板块的开头。

参考信息：Footer-Fixed-Lines 属性

“Common”选项卡

属性A	XML 属性名称	描述
忽略行 - 正则表达式	ignore-lines-regexp	一个正则表达式，用于从页脚区块中排除匹配的行。
行计数	lines-count	页脚区块占用的行数。
剪裁尾部忽略行	trim-eof-ignore-lines	确定是否从页脚中去除正则表达式的匹配行。

参考信息：Footer-Starts-With 属性

“Common”选项卡

属性A	XML 属性名称	描述
正则表达式	regexp	一个正则表达式，用于标识数据记录内容板块的开头。

Aggregator 构成子元素

概念：Aggregator 构成子元素

“aggregator”构成支持以下子元素。

子元素	描述
custom-batch-strategy	您配置为 Spring Bean 的可自定义的批处理策略。
size-batch-strategy	根据指定的批处理大小值对传入消息进行批处理。
time-batch-strategy	根据指定的时段值对传入消息进行批处理。
custom-collater	您配置为 Spring Bean 的可自定义的整理策略。
message-content-collater	聚合的消息内容 MediationMessage。
xml-message-content-collater	聚合 XML 消息流的 XML 文档内容。
json-collater	将多个 JSON 文档聚合到一个输出中。
zip-file-collater	将多个文件聚合到一个 ZIP 文件中。
mtable-collater	聚合 mtable 行合并到新表格中 mtable 实例。

参考信息：Custom-Batch-Strategy 属性

“Common”选项卡

属性A	XML 属性名称	描述
java - java - java	ref	用于实施 custom-batch-strategy 接口类的 Spingle。

参考信息：Size-Batch-Strategy 属性

“Common”选项卡

属性A	XML 属性名称	描述
批处理大小	batch-size	一个批处理可串联的最大消息数。

参考信息：Time-Batch-Strategy 属性

“Common”选项卡

属性A	XML 属性名称	描述
时段	time-period	批处理聚合的持续时间。
时间单位	time-unit	“时段”属性实施的时间单位。默认为“seconds”。

参考信息：Custom-Collater 属性

“Common”选项卡

属性A	XML 属性名称	描述
输出	output	指定 custom-collater 子元素将输出定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。 <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。

属性A	XML 属性名称	描述
		变量：将输出另存为中的已命名变量 <code>MediationContext</code>。
java - java - java	ref	用于实施 <code>custom-collater</code> 接口类的 <code>Spingle</code> 。

参考信息：Message-Content-Collater 属性

“Common”选项卡

属性A	XML 属性名称	描述
输出	output	指定 <code>message-content-collater</code> 子元素将输出定向到何处。共有 5 个可能的值： <code>message</code>：整条消息，包括 <code>rootpart</code> 和附件。此值会创建一条新消息，同时会移除所有附件。 <code>soapbody</code>：中 SOAP 正文的内容 <code>rootpart</code> 消息的“”。 <code>attachment</code>：消息的 MIME 附件。指定附件索引。 <code>rootpart</code>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 <code>MediationContext</code>。
标头文本	header-text	向输出消息添加标头。
Footer Text		在输出消息中添加尾部。
分隔符	separator	在批处理中连接消息时使用的字符串。

参考信息：XML-Message-Content-Collater 属性

“Common”选项卡

属性A	XML 属性名称	描述
输出	output	指定 <code>xml-message-content-collater</code> 子元素将输出定向到何处。共有 5 个可能的值： <code>message</code>：整条消息，包括 <code>rootpart</code> 和附件。此值会创建一条新消息，同时会移除所有附件。

属性A	XML 属性名称	描述
		<i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 <i>attachment</i>：消息的 MIME 附件。指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 <i>MediationContext</i>。
标头文本	header-text	向输出消息添加标头。
Footer Text		在输出消息中添加尾部。
分隔符	separator	在批处理中连接消息时使用的字符串。
命名空间	namespaces	XPath 表达式使用的命名空间。
XPath	xpath	一个 XPath 表达式，用于聚合匹配的 XML 消息。如果未指定，则“xml-message-content-collater”会将所有 XML 消息内容汇集到一个输出文件中。

参考信息：JSON-Aggregator 属性

“Common”选项卡

属性A	XML 属性名称	描述
输出	output	指定 json-collater 子元素将输出定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 <i>attachment</i>：消息的 MIME 附件。指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 <i>MediationContext</i>。

参考信息：Zip-File-Collater 属性

“Common”选项卡

属性A	XML 属性名称	描述
输出	output	指定 zip-file-collater 子元素将输出定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。 <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 MediationContext。
格式	format	输出文件的格式。有以下选项： - zip - tar
消息实体名称	message-entity-name	添加到输出文件的每条消息的名称。

参考信息：Mtable-Collater 属性

“Common”选项卡

属性A	XML 属性名称	描述
输出	output	Workday 会忽略 mtable-collater 子元素的此设置。
表名称	mtable-name	的名称 MediationContext 用于存储输出的属性 mtable。
行名称	row-name	的名称 MediationContext 属性（用于存储您的聚合数据）mtable 行
列名称	column-names	用于定义输出的以逗号分隔的列表 mtable。
索引列	index-column	输出的索引列名称 mtable。

属性A	XML 属性名称	描述
允许 null 行	allow-null-row	确定您的输出是否 mtable 支持 null 行值。

通用构成

概念：通用构成

Workday Studio 提供一系列可用于构建集成的预封装子程序集。这些部件装配件可从“面板”的“通用组件”选项卡获取。当您使用 local-out 将 XML 源中的元素与 Workday 中的相应端点关联。

它还显示可用于该子部件的参数，以及已选择的必需参数。您可以随时在子程序集的“Properties”视图中调整参数。

Workday Studio 在“Palette”中包含了以下通用构成：

名称	函数	注释
Ftp	路由至“ftp-out”传输。	- 为所有 FTP 传输提供单个入口点。 - 使用 <code>wd.ftp.endpoint</code> 参数值来确定实施的 FTP 传输类型。示例： <code>ftp://</code>，<code>sftp://</code>，或 <code>ftps://</code>。 - 显示 Workday 中常用的传输设置，例如用户名和密码身份验证。隐藏更高级的公钥身份验证。 注: Studio 调色板中不再提供“FTP 子装配件”选项。在新集成中，您应使用 <code>ftp-out</code> 或 <code>sftp-out</code> 组件将文档发送到 FTP 服务器。这些组件可提高安全性、性能和可靠性。它们还与更高版本的 JSCAPE 兼容。该 FTP 子程序集在现有集成中仍然可用，但 Workday 建议您在切实可行的情况下将其替换。
GetEventConfigurations	检索附加到特定集成事件的服务配置。	
GetEventDocuments	检索附加到特定集成事件的文档。	
GetIntegrationEvent	根据 Workday 参考编号检索集成事件。	
GetIntegrationSystems	从 Workday 中检索集成系统的配置并进行解析。	
PagedGet	实现任何所需的分页逻辑。 <code>paged-getWeb</code> 服务操作，支持逐页检索数据。	使用 <code>is.paged.get.process.endpoint</code> 参数可将每一页响应数据

名称	函数	注释
		发送到另一个子部件进行处理。 - 使用 <code>is.paged.get.aggregate.header</code> , <code>is.paged.get.aggregate.footer</code> 和 <code>is.paged.get.aggregate.xpath</code> 参数用于汇总所有页面的回复数据。 - 聚合的 XML 数据可用作“”中的消息。 <code>MediationContext</code> 子装配件返回时间。 - 如果 Web 服务预计将返回大量页面，请使用逐页处理方法。
PagedGetLocalPaging	实现任何所需的分页逻辑。 <code>paged-get</code> 本地 Web 服务操作，支持逐页检索数据。	- 允许在 ESB 服务器本地执行分页流程，而不是通过 Workday 远程执行。 - 使用单个查询来返回所有 Workday 编号。后续调用会检索每个 Workday 编号的数据。
PdfPrintStep	将自定义报告数据从 <code>workday-out-rest</code> 使用 BIRT 设计将请求转换为 PDF 格式的报告。	用单引号将字符串值括起来，从而为每个支持 MVEL 的参数指定一个字符串值。
PutIntegrationEvent	在 Workday 中更新集成系统的集成事件状态。	- 使用 <code>Put_Integration_Event</code> Web 服务操作。 - 将从 Web 服务操作返回的事件响应 Workday 编号 <code>MediationContext</code> 属性 <code>integrations.event.response.wi</code>
PutIntegrationMessage	向 Workday 发送更新集成消息。	- 还可以将引用形式的文档添加到 Workday 文档存储中。 - 使用 <code>Put_Integration_Message</code> Web 服务操作。 - Workday 将 <code>PutIntegrationMessage</code> 子程序集在单个集成中可以进行的调用次数限制为 500 次。后续集成消息将记录到日志文件中，而且无法在集成事件中查看（重定向不适用于具有附件或会更改集成事件状态的消息）。

名称	函数	注释
SalesforceConnector	将经过身份验证的消息发送到 salesforce.com。	
PrismConnector	将 CSV 数据发送到 Prism Analytics。	

参考信息：FTP 子程序集属性

公用选项卡

注：Studio 调色板中不再提供“FTP 子装配件”选项。在新集成中，您应使用 ftp-out 或 sftp-out 组件将文档发送到 FTP 服务器。这些组件可提高安全性、性能和可靠性。它们还与更高版本的 JSCAPE 兼容。该 FTP 子程序集在现有集成中仍然可用，但 Workday 建议您在切实可行的情况下将其替换。

属性A	XML 属性名称	描述
编号	id	Ftp 子装配件在程序集中的唯一编号。
将回复发送至	routes-response-to	Ftp 子程序集响应消息的目标。
执行时间	execute-when	确定是否执行 Ftp 子程序集的条件。
端点	endpoint	子装配件的目标端点地址。Ftp 组件支持 vm:// 端点地址。示例：vm://wcc/Ftp。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	clone-request	指定是否在子装配件之间进行链式处理。设置为“true”可创建原始消息的副本，这样任何修改请求消息的子程序集都不会影响其他

属性A	XML 属性名称	描述
		子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配在标记 MediationException 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
wd.ftp.endpoint	此端点方案用于确定要使用的 FTP 输出 传输方式。端点必须以以下开头：ftp://，sftp://，或 ftps://。
wd.ftp.username	用于访问 FTP 服务器的用户名。
wd.ftp.password	与用于访问 FTP 服务器的用户名关联的密码。
wd.ftp.method	与端点连接时，FTP 子部件实施的方法或消息操作。请将每项操作单引号引起来。有效选项包括： 'put'：FTP 组件将消息发送到 FTP 服务器，并且消息保持不变。 'get'：程序集会将消息的根部分替换为从 FTP 服务器接收的数据。 'list'：消息保持不变。Workday 添加了一个名为“”的上下文属性 wd.ftp.files.list，其中包含 List<String> 对象实例。 注：使用 'list' 方法，文件模式为 *.txt 或 *.xml 会导致异常。 'delete'：Workday 请求从 FTP 服务器删除文件。该消息保持不变。 'mget' (MultiGet)：Workday 下载与输入模式匹配的所有文件，并将消息的根目录部分替换为包含这些文件的 ZIP 文件。 'mdelete' (MultiDelete)：Workday 会删除与输入模式匹配的所有文件。该消息保持不变。
wd.ftp.passive.mode	指定在 FTP 连接期间是否使用被动模式。
wd.ftp.file.pattern	指定执行 FTP 操作时使用的文件模式。示例： *.xml *.txt

参数	描述
	对于除 list, Workday 支持通配符，并在客户端执行筛选。对于 list 方法，Workday 使用基于正则表达式语法的服务器端筛选。示例： - wd.ftp.method 是 'get' 与 wd.ftp.file.pattern 是 *.xml：Workday 会获取在 FTP 服务器上的指定目录中找到的第一个 .xml 文件。 - wd.ftp.method 是 'mget' 与 wd.ftp.file.pattern 是 *.xml：Workday 会获取在 FTP 服务器的指定目录下找到的所有 .xml 文件，并以 ZIP 格式返回这些文件。 - wd.ftp.method 是 'list' 与 wd.ftp.file.pattern 是 ^.*\.(xml)\$：Workday 会列出在 FTP 服务器的指定目录中找到的所有 .xml 文件。
wd.ftp.temp.file.name	指定在执行 FTP 操作时是否应使用临时文件。put 消息操作。
wd.ftp.timeout	消息操作的超时设置（以毫秒为单位）。
wd.ftp.directory	相对路径或完全限定的目录路径。
wd.ftp.private.key	与用户名关联的私钥，用于 SFTP 身份验证。使用附件协议和 Workday 编号，将其设置为 URI 格式。示例：attachment:privatekey/@{WID}，其中 Workday 编号对应于 32 个字符的 Workday 编号。
wd.ftp.max.attempts	访问 FTP 服务器的最大尝试次数。
wd.ftp.debug	指定 Workday 是否向日志添加调试日志记录。
wd.ftp.proxy.host	FTP 代理服务器的主机名。
wd.ftp.proxy.port	FTP 代理服务器端口。
wd.ftp.proxy.username	用于访问 FTP 代理服务器的用户名。
wd.ftp.proxy.password	FTP 代理服务器的密码。
wd.ftp.proxy.type	代理服务器的类型。
wd.ftp.input.content.type	指定使用 FTP 检索的文件的内容类型 get。示例：text/plain; charset=Big5。

参考信息：GetEventConfigurations 子程序集属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中的 GetEventConfigurations 子程序集的唯一编号。

属性A	XML 属性名称	描述
将回复发送至	routes-response-to	GetEventConfigurations 子部件的响应消息的目的地。
执行时间	execute-when	一个用于确定是否执行 GetEventConfigurations 子程序集的条件。
端点	endpoint	指定子装配件的目标端点地址。GetEventConfigurations 组件支持 vm:// 端点地址。示例：vm://wcc/GetEventConfigurations。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	clone-request	指定是否在子装配件之间进行链式处理。设置为“true”可创建原始消息的副本，这样任何修改请求消息的子程序集都不会影响其他子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配件在标记 MediationContext 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
<code>ie.event.wid</code>	您要检索其服务配置的集成事件的 Workday 编号。自动填充为 <code>lp.isSet() ? lp.getIntegrationEventWID() : null</code> 。
<code>is.event.disable.msg.storage</code>	设置为“”时关闭消息存储 <code>true</code> 。
<code>ie.event.wws.version</code>	集成事件 Web 服务版本。

参考信息：GetEventDocuments 子程序集属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	<code>id</code>	程序集中的 GetEventDocuments 子程序集的唯一编号。
将回复发送至	<code>routes-response-to</code>	GetEventDocuments 子程序集的响应消息的目标。
执行时间	<code>execute-when</code>	一个用于确定是否执行 GetEventDocuments 子程序集的条件。
端点	<code>endpoint</code>	子装配的目标端点地址。GetEventDocuments 组件支持 <code>vm://</code> 端点地址。示例： <code>vm://wcc/GetEventDocuments</code> 。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	<code>store-message</code>	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	<code>transport-class</code>	如果需要，为每个装配指定一个备选的实现类。 您可以通过编辑 <code>WEB-INF/classes/spring/assembly-bean.xml</code> 文件来全局更改实现类。
忽略动态端点	<code>ignore-dynamic-endpoints</code>	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。

属性A	XML 属性名称	描述
		使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	clone-request	指定是否在子装配件之间进行链式处理。设置为“true”可创建原始消息的副本，这样任何修改请求消息的子程序集都不会影响其他子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配件在标记 MediationException 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
ie.event.wid	您要检索其服务配置的集成事件的 Workday 编号。自动填充为 lp.isSet() ? lp.getIntegrationEventWID() : null。这意味着，如果已在中介上下文中指定启动文档，则该参数的值默认为集成事件 Workday 编号，否则默认为 null。

参考信息：GetIntegrationEvent 子程序集属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中的 GetIntegrationEvent 子程序集的唯一编号。
将回复发送至	routes-response-to	GetIntegrationEvent 子程序集的响应消息的目标。
执行时间	execute-when	用于确定是否执行 GetIntegrationEvent 子程序集的条件。
端点	endpoint	子装配件的目标端点地址。GetIntegrationEvent 组件支持 vm:// 端点地址。示例：vm://wcc/GetIntegrationEvent。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	clone-request	指定是否在子装配件之间进行链式处理。设置为“true”可创建原始消息的副本，这样任何修改请求消息的子程序集都不会影响其他子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配件在标记 MediationContext 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
ie.event.wid	您要检索其服务配置的集成事件的 Workday 编号。

参考信息：GetIntegrationSystems 子程序集属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中的 GetIntegrationSystems 子程序集的唯一编号。
将回复发送至	routes-response-to	GetIntegrationSystems 子程序集的响应消息的目标。
执行时间	execute-when	一个用于确定是否执行 GetIntegrationSystems 子程序集的条件。
端点	endpoint	子装配的目标端点地址。GetIntegrationSystems 组件支持 vm:// 端点地址。示例：vm://wcc/GetIntegrationSystems。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	clone-request	指定是否在子装配之间进行链式处理。设置为“true”可创建原始消息的副本，这样任何修改请求消息的子程序集都不会影响其他子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。

属性A	XML 属性名称	描述
		子装配件在标记 MediationContext 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
is.system.wid	您要检索和解析其服务配置的集成系统的 Workday 编号。自动填充为 lp.getIntegrationSystemRefWID()。

参考信息：PagedGet 子程序集属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	程序集中的 PagedGet 子程序集的唯一编号。
将回复发送至	routes-response-to	PagedGet 子部件的响应消息的目标。
执行时间	execute-when	一个用于确定是否执行 PagedGet 子程序集的条件。
端点	endpoint	子装配件的目标端点地址。PagedGet 组件支持 vm://端点地址。示例：vm://wcc/PagedGet。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA)

属性A	XML 属性名称	描述
		值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	clone-request	指定是否在子装配件之间进行链式处理。设置为“true”可创建原始消息的副本，这样任何修改请求消息的子程序集都不会影响其他子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配件在标记 MediationContext 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
is.paged.get.request.current.page.xpath	请求文档中的 XPath 位置 PagedGet 组件用于设置正在请求的页数。
is.paged.get.response.current.page.xpath	用于从 Web 服务响应消息中提取当前页面值的 XPath 表达式。
is.paged.get.response.total.pages.xpath	XPath 表达式，用于从 Web 服务响应消息中提取总页数的值。
is.paged.get.response.total.results.xpath	个 XPath 表达式，用于从 Web 服务响应消息中提取“总结果页”的值。
is.paged.get.process.endpoint	处理每页结果的子程序集的 本地输入 端点。示例：vm://payrollInterface/ProcessPayeesLocalIn。 注：如果您设置此参数，Workday 将忽略聚合参数。 这些 MediationContext 属性在处理子部件中可用： - is.paged.get.request.current.page.xpath - is.paged.get.response.current.page.xpath - is.paged.get.response.total.pages.xpath - is.paged.get.response.total.results.xpath
is.paged.get.aggregate.header	XML 片段形式的聚合响应的标头值。示例： <AllData>。

参数	描述
<code>is.paged.get.aggregate.footer</code>	XML 片段形式的聚合响应的标头值。示例： <code></AllData></code> 。
<code>is.paged.get.aggregate.xpath</code>	聚合时应用于响应消息的 XPath 表达式。
<code>is.paged.get.namespaces</code>	指定的命名空间在文档中不可用。
<code>is.paged.get.application</code>	您要调用的 Web 服务的应用程序组。示例： <code>Human_Resources</code> 。
<code>is.paged.get.version</code>	您要调用的公共 Web 服务的版本。最佳做法是，在构建集成时使用最新的 Workday Web 服务 (WWS) 版本。请勿使用会随着集成运行而变化的动态最新版本。 对于版本早于 2016.45 的程序集，此参数将自动填充为 <code>util.assemblyVersionAsWWSVersion</code>。 对于版本为 2016.45 或更高版本的程序集，必须明确提供 Web 服务版本。示例： <code>v27.2</code>。
<code>is.paged.get.page.zero</code>	指定在总页数为零的情况下是否应生成回复页。
<code>is.paged.get.parallel</code>	指定是否并行检索 Web 服务响应，从而使中指定的组件能够并行运行。 <code>is.paged.get.process.endpoint</code> 参数来处理每个寻呼响应。自动填充为 <code>false</code>，这表示系统会聚合 Web 服务响应，而不是并行检索这些响应。 Workday 使用独特的上下文处理并行响应，这些上下文允许对在输入 PagedGet 之前设置的属性值和变量进行只读访问。子装配件中每一页的处理完成后，系统都会丢失该处理期间设置的任何属性或变量。PagedGet 仍会设置 <code>is.paged.get.total.pages</code> 与 <code>is.paged.get.total.results</code> 调用方的属性 <code>MediationContext</code>。该 <code>is.paged.get.current.page</code> 在并行分页上下文中设置了属性。 Workday 在同一相关信息中处理非并行响应，并处理子装配件中的页面。这意味着您可以在子部件中设置属性和变量，并检查先前页面的状态。
<code>is.paged.get.store.requests</code>	指定是否在消息存储中存储每个数据页的 Web 服务请求消息。
<code>is.paged.get.overall.timeout.seconds</code>	整体超时时间（秒）。如果您未设置，默认值为 6 小时。您可以设置的最长超时时间为 30 小时。
<code>is.paged.get.timeout.seconds</code>	处理每个页面的超时时间。如果您未设置，默认值为 6 小时。不能超过总体超时值。

“Out Parameters”选项卡

在“PagedGet”子程序集执行处理时，它会传递以下输出参数值，以供下游构成使用：

输出参数	描述
<code>is.paged.get.last.page</code>	Web 服务操作返回的最终页面。
<code>is.paged.get.current.page</code>	当前正在处理的页码。
<code>is.paged.get.total.pages</code>	要处理的总页数。
<code>is.paged.get.total.results</code>	Web 服务操作返回的结果总数。

参考信息：PagedGetLocalPaging 子程序集属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	<code>id</code>	程序集中的 PagedGetLocalPaging 子程序集的唯一编号。
将回复发送至	<code>routes-response-to</code>	PagedGetLocalPaging 子程序集的响应消息的目标。
执行时间	<code>execute-when</code>	一个用于确定是否执行 PagedGetLocalPaging 子程序集的条件。
端点	<code>endpoint</code>	子装配件的目标端点地址。PagedGetLocalPaging 组件支持 <code>vm://</code> 端点地址。示例： <code>vm://wcc/PagedGet</code> 。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	<code>store-message</code>	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	<code>transport-class</code>	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 <code>WEB-INF/classes/spring/assembly-bean.xml</code> 文件来全局更改实现类。
忽略动态端点	<code>ignore-dynamic-endpoints</code>	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	<code>clone-request</code>	指定是否在子装配件之间进行链式处理。设置为“true”可创建原始

属性A	XML 属性名称	描述
		消息的副本，这样任何修改请求消息的子程序集都不会影响其他子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配在标记 MediationContext 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
is.paged.get.process.endpoint	指定将处理每页结果的子程序集的 local-in 端点。示例：vm://payrollInterface/ProcessPayeesLocalIn。 注：如果您设置此参数，Workday 将忽略聚合参数。 这些 MediationContext 属性在处理子部件中可用： - is.paged.get.request.current.page.xpath - is.paged.get.response.current.page.xpath - is.paged.get.response.total.pages.xpath - is.paged.get.response.total.results.xpath
is.paged.get.page.size	每页数据中要检索的记录数。自动填充为 100。
is.paged.get.response.reference.wids.xpath	用于在“Get all Workday ID”调用响应消息中查找 Workday 编号的 XPath 表达式。示例：//wd:Payee_Reference/wd:ID[@wd:type='WID']。
is.paged.get.request.reference.wids.parent.xpath	此 XPath 表达式用于查找请求消息的父级元素，在获取每页数据时将指示 Workday 编号。
is.paged.get.request.reference.wids.elementName	要在每页数据的请求中使用的 Workday 编号的元素名称。
is.paged.get.application	您要调用的 Web 服务的应用程序组。示例：Human_Resources。
is.paged.get.version	您要调用的公共 Web 服务的版本。最佳做法是，在构建集成时使用最新的 Workday Web 服务 (WWS) 版本。请勿使用会随着集成运行而变化的动态最新版本。 对于版本早于 2016.45 的程序集，此参数将自动填充为 util.assemblyVersionAsWWSVersion。

参数	描述
	对于版本为 2016.45 或更高版本的程序集，必须明确提供 Web 服务版本。示例：v27.2。
is.paged.get.xpath.ns.prefix	在所有 XPath 表达式中使用的 Workday XML 命名空间的前缀。自动填充为 wd。

参考信息：PdfPrintStep 子程序集属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	部件内的 PdfPrintStep 子部件的唯一编号。
将回复发送至	routes-response-to	PdfPrintStep 子部件的响应消息的目标。
执行时间	execute-when	确定是否执行 PdfPrintStep 子程序集的条件。
端点	endpoint	子装配的目标端点地址。PdfPrintStep 组件支持 vm:// 端点地址。示例：vm://wcc/PdfPrintStep。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	clone-request	指定是否在子装配之间进行链式处理。设置为“true”可创建原始消息的副本，这样任何修改请求消息的子程序集都不会影响其他

属性A	XML 属性名称	描述
		子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配在标记 MediationException 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
pdf.report.design.variable	包含报告设计的集成变量的名称，用于定义打印的自定义报告的布局。示例： 'test.report.design'。 报告设计定义包含在扩展名为“.rptdesign”的文件中。
pdf.workday.report.variable	包含原始 Workday 报告数据的集成变量的名称。示例：'test.workday.report'。
pdf.timezone	Java 时区（用于设置格式）示例： 'US/Eastern' 'GMT' 'CET'
pdf.language	报告的语言，用于格式设置。示例： 'en_US' 'fr_FR' 'de_DE'
pdf.locale	报告的区域设置（用于设置格式）示例： 'en_US' 'fr_FR' 'de_DE'
pdf.business.form.layout.wid	正在使用的业务表单布局的 Workday 编号。允许检索在布局上定义为令牌的图片。
pdf.report.design.wid	正在使用的报告设计的 Workday 编号。启用翻译。
pdf.apply.report.design.per.row	指定 Workday 是将报告设计应用于单个报告行，还是将报告作为一个整体。

relatedlinks

参考信息

[The Next Level: Using BIRT in Studio Integrations](#)

参考信息：PutIntegrationEvent 子程序集属性

“Common”表格

属性	XML 属性名称	描述
编号	id	子程序集 “PutIntegrationEvent” 在程序集中的唯一编号。
将回复发送至	routes-response-to	PutIntegrationEvent 子程序集的响应消息的目标。
执行时间	execute-when	一个条件，用于确定是否执行 PutIntegrationEvent 子程序集。
端点	endpoint	子装配件的目标端点地址。PutIntegrationEvent 组件支持 vm:// 端点地址。示例：vm://wcc/PutIntegrationEvent。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	clone-request	指定是否在子装配件之间进行链式处理。设置为“true”可创建原始消息的副本，这样任何修改请求消息的子程序集都不会影响其他子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配件在标记 MediationContext 状态为“已中止”。

属性A	XML 属性名称	描述
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
<code>is.event.wid</code>	Workday 要将消息关联到的集成事件的 Workday 编号。 如果您未指定此值，并且该值不在 <code>lp MVEL</code> 变量，则 Workday 会使用 <code>is.system.wid</code> 改为使用参数，将消息链接到集成系统。自动填充为 <code>lp.isSet() ? lp.getIntegrationEventWID() : null</code> 。
<code>is.system.wid</code>	集成系统的 Workday 编号。 自动填充为 <code>lp.isSet() ? lp.getIntegrationSystemRefWID() : null</code> 。 仅当集成事件编号不可用时，Workday 才会使用此参数。
<code>is.process.desc</code>	集成事件描述。Workday 会在其流程监视器中显示此值。在创建新的集成事件时为必填项。
<code>is.event.initiated</code>	集成事件发起的日期和时间。如果您输入字符串值 'current'，当前日期和时间将发送到 Workday。
<code>is.event.completed</code>	集成事件完成的日期和时间。如果您输入字符串值 'current'，当前日期和时间将发送到 Workday。
<code>is.response.msg</code>	集成事件的最新消息或活动自由格式的文本字段。
<code>is.event.member.wids</code>	以逗号分隔的 Workday 编号列表，该列表表示关联到此集成事件的成员。
<code>is.percent.complete</code>	Workday 用于维护“完成百分比”进度条的值。该值可以是可转换为 Java BigDecimal 对象的任何类型。该值必须介于 0 到 100 之间。

参考信息：PutIntegrationMessage 子程序集属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	id	子程序集 “PutIntegrationMessage”在程序集中的唯一编号。
将回复发送至	routes-response-to	PutIntegrationMessage 子程序集的响应消息的目标。

属性A	XML 属性名称	描述
执行时间	execute-when	一个条件，用于确定是否执行 PutIntegrationMessage 子程序集。
端点	endpoint	子装配件的目标端点地址。PutIntegrationMessage 组件支持 vm:// 端点地址。示例：vm://wcc/PutIntegrationMessage。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	store-message	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	transport-class	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 WEB-INF/classes/spring/assembly-bean.xml 文件来全局更改实现类。
忽略动态端点	ignore-dynamic-endpoints	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	clone-request	指定是否在子装配件之间进行链式处理。设置为“true”可创建原始消息的副本，这样任何修改请求消息的子程序集都不会影响其他子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配件在标记 MediationContext 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
<code>is.event.wid</code>	Workday 要将消息关联到的集成事件的 Workday 编号。 如果您未指定此值，并且该值不在 <code>lp MVEL</code> 变量，则 Workday 会使用 <code>is.system.wid</code> 改为使用参数，将消息链接到集成系统。自动填充为 <code>lp.isSet()</code> ? <code>lp.getIntegrationEventWID()</code> : <code>null</code> 。
<code>is.system.wid</code>	集成系统的 Workday 编号。 自动填充为 <code>lp.isSet()</code> ? <code>lp.getIntegrationSystemRefWID()</code> : <code>null</code> 。 仅当集成事件编号不可用时，Workday 才会使用此参数。
<code>is.message.severity</code>	指定消息的严重程度：'CRITICAL'，'DEBUG'，'ERROR'，'INFO'，或 'WARNING'。自动填充为 'INFO'。
<code>is.message.summary</code>	发送到 Workday 的消息汇总。
<code>is.message.detail</code>	发送到 Workday 的消息详细信息。
<code>is.message.detail.richtext</code>	确定消息详细信息是否设置为富文本。自动填充为 <code>true</code> 。
<code>is.document.variable.name</code>	中变量的名称 <code>MediationContext</code> 包含程序集存储步骤的 <code>Acom</code> 架构输出。可变内容用于添加对集成事件的引用。 该 <code>store</code> 用于创建文档的步骤必须将架构属性设置如下： <code>schema="http://www.w3.org/2005/Atom"</code> 。
<code>is.document.file.name</code>	文档的文件名。如果您不提供标题，Workday 将使用 <code>Atomic</code> 文档中的文档标题。
<code>is.document.owner.wid</code>	文档所有者的 Workday 编号。如果您不提供，Workday 将使用当前集成系统用户的编号。
<code>is.document.labels</code>	当文档在集成事件编号或关联的业务流程的范围内时，应用于文档的逗号分隔标签列表。
<code>is.document.deliverable</code>	指定是否将添加到集成事件的文档自动传送给客户或外部提供商。自动填充为 <code>'true'</code> 。
<code>is.document.delivery.services</code>	以逗号分隔的集成服务名称列表，用于标记要附加到集成事件的文档。默认情况下，Workday 会使用与集成系统关联的所有传送服务。
<code>is.document.retrieved</code>	指定是否将文档视为已由检索服务检索到。自动填充为 <code>'false'</code> 。
<code>is.message.targets</code>	此消息的目标对象的逗号分隔引用列表。这些目标将成为 Workday 应用程序的“View Integration Message”选项卡中的对象链接。
<code>is.message.targets.type</code>	目标人员参考类型。自动填充为 <code>'WID'</code> 。

参数	描述
<code>is.message.secured.instance.refs</code>	此消息的受保护实例引用列表（逗号分隔）。
<code>is.message.secured.instance.refs.type</code>	允许重设受保护实例引用的默认类型。自动填充为 'WID'。
<code>is.message.storage.enabled</code>	确定是否存储该消息。自动填充为 'false'。

参考信息：SalesforceConnector 子程序集属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	<code>id</code>	程序集中 SalesforceConnector 子程序集的唯一编号。
将回复发送至	<code>routes-response-to</code>	SalesforceConnector 子部件的响应消息目标。
执行时间	<code>execute-when</code>	一个条件，用于确定是否执行 SalesforceConnector 子程序集。
端点	<code>endpoint</code>	子装配件的目标端点地址。SalesforceConnector 组件支持 <code>vm://</code> 端点地址。示例： <code>vm://wcc/SalesforceConnector</code> 。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	<code>store-message</code>	除非 Workday 技术支持另行通知，否则无指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	<code>transport-class</code>	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 <code>WEB-INF/classes/spring/assembly-bean.xml</code> 文件来全局更改实现类。
忽略动态端点	<code>ignore-dynamic-endpoints</code>	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。 使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	<code>clone-request</code>	指定是否在子装配件之间进行链式处理。设置为“true”可创建原始消息的副本，这样任何修改请求

属性A	XML 属性名称	描述
		消息的子程序集都不会影响其他子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配在标记 MediationContext 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
wd.sf.auth.username	您的 salesforce.com 开发者账户用户名
wd.sf.auth.password	您的 salesforce.com 开发者账户密码。
wd.sf.msg.varname	Workday 发送到 salesforce.com 端点的 XML 消息。
wd.sf.auth.token	您的 salesforce.com 开发者帐户安全令牌。
wd.sf.auth.endpoint	可选的 salesforce.com 端点。默认端点已配置为子部件号中的一部分。 http-out 运输
wd.sf.api.custom.version	可选的 salesforce.com API 版本。如果您未提供值，则使用端点 API 版本。

PrismAnalytics 子组件

步骤：将 **CSV** 数据发送到 **Prism Analytics**

开始之前

安全性：“System”功能区域中的“Security Administration”域

关于此任务

利用 Workday Studio 的“PrismAnalytics”构成，您可以将 CSV 数据发送到 Workday Prism Analytics。在将程序集配置为使用“PrismAnalytics”之前，您必须在 Workday 中注册 Prism API 客户端并获取 REST API 端点。您还必须为源 CSV 数据创建 JSON 描述。

过程

1. [注册 Workday Prism Analytics API 客户端](#) on page 104.
在 Workday 中，注册集成 API 客户端，并以 Prism Analytics 为其范围。获取 PrismAnalytics 组件需要用作参数的 客户端 ID、客户端密码和 刷新令牌 的值。
2. [获取 Workday REST API 端点](#) on page 104.
在 Workday 中，获取 PrismAnalytics 组件所需的 Workday REST API 端点（用作参数）。
3. [描述源 CSV 数据](#) on page 105.
使用 JSON 文本描述源 CSV 数据，以便作为变量传递到 PrismAnalytics 组件。

4. [配置具有 PrismAnalytics 子程序集的程序集](#) on page 110.
- 在 Workday Studio 中，创建一个程序集，用于获取 CSV 数据并将数据传递给正确配置的 PrismAnalytics 组件。

relatedlinks
参考信息

[Workday 31 What's New Post: Workday Studio](#)

注册 **Workday Prism Analytics API** 客户端

开始之前

安全性：“System”功能区域中的“Security Administration”域

关于此任务

要使用 Workday Studio 的“PrismAnalytics”构成，您必须在 Workday 中注册一个 API 客户端，并以 Prism Analytics 作为其范围。该构成需要 Workday 在注册过程中生成的客户端 ID 值和客户端密码值。它还需要一个刷新令牌值。

过程

1. 访问“Register API Client for Integrations”任务。完成此任务时，请参考以下信息：

选项	描述
Non-Expiring Refresh Tokens	防止刷新令牌超时。由 Workday 自动选择。请勿清除。
Scope (Functional Area)	选择“Prism Analytics”。
Include Workday Owned Scope	提供对 Workday 核心域的访问权限，这些域不在任何功能区域中。选中此复选框。

2. 复制“客户端 ID”和“客户端密码”的值。Studio 的 Prism Analytics 组件在运行时需要使用这些组件。
wd.prism.component.client.id 与 wd.prism.component.client.secret 参数
3. 从相关操作菜单中，选择“API 客户端”>“管理集成的刷新令牌”。
4. 选择一个要与刷新令牌关联的 Workday 账户。该账户必须有权在 Workday 中创建 Prism 表格。
注：您无法以集成系统用户的身份执行这些步骤。
5. 选择“Generate New Refresh Token”。
6. 复制“刷新令牌”值。PrismAnalytics 组件的应用需要该组件。
wd.prism.component.refresh.token 参数。

获取 **Workday REST API** 端点

开始之前

安全性：“System”功能区域中的“Security Administration”域

关于此任务

Workday Studio 的“PrismAnalytics”构成需要 Workday REST API 端点。您可在 Workday 中找到此端点。

过程

1. 访问“View API Clients”报告。
2. 复制 Workday REST API 端点 值。
示例：<https://i-07d55d20e81e2d191.workdaysuv.com/ccx/api/v1/super>。

- 删除 `https://` 位于开头，并且 `/ccx/api/v1/<tenant_name>` “”
示例：`i-07d55d20e81e2d191.workdaysuv.com。`
- 复制文本。Studio 的 Prism Analytics 组件需要使用该组件 `wd.prism.component.api.hostname` 参数。

描述源 CSV 数据

关于此任务

Workday Studio 的“PrismAnalytics”构成需要获得源 CSV 数据的描述。您可以在所选工具中创建 JSON 文本，然后在 Workday Studio 中进行使用。示例：将其粘贴到“write”步骤的“Message Builder”选项卡上的文本消息中。

过程

- 创建包含 2 个主节点的 JSON 文本：
 - `parseOptions`：有关 CSV 数据格式的信息，包括字符集和分隔符字符。
 - `fields`：CSV 数据中每个字段的相关信息，包括数据类型和格式。
- 在 `parseOptions` 节点时，您可以定义以下字段：

字段名称	类型	必需	描述
<code>charsetName</code>	对象	是	CSV 数据文件的字符集。示例： <pre>"charset": { "id": "Encoding=UTF-8" }</pre> 目前，Workday 仅支持 UTF-8 编码格式。Workday 最终以 Unicode 格式存储数据。没有 Unicode 等效项的字符将存储为 Unicode 替换字符，即一个黑色菱形背景下的问号。
<code>fieldsEnclosedBy</code>	String	否	用于将 CSV 数据中的字段括起来的字符。默认为英文双引号。如果在字段中使用了双引号，请在前面添加另一个双引号以将其转义。示例： <pre>"fieldsEnclosedBy": "\\""</pre>

字段名称	类型	必需	描述
fieldsDelimitedBy	String	是	用于分隔 CSV 数据中字段值的字符。示例： <pre>"fieldsDelimitedBy": ","</pre>
headerLinesToIgnore	数字	否	解析器要忽略的标头行数。设置为 0 如果 CSV 数据没有标头示例： <pre>"headerLinesToIgnore": 1</pre>
type	对象	是	输入文件的类型。目前，Workday 仅支持分隔文件。示例： <pre>"type": { "id": "Schema_File_Type=Delimit</pre> 注：此值区分大小写。请注意“带分隔符”字段中的大写 D。

3. 在 fields 节点时，您可以定义以下字段：

字段名称	类型	必需	描述
ordinal	数字	是	字段在 CSV 数据集中的顺序（即索引）。序数值： • 必须从 1 开始。 • 必须是连续的。 • 不能跳过任何数字。 • 最多 1000 个。
name	String	是	CSV 数据集字段的名称。字段名称： • 在数据集中必须唯一。 • 最多可包含 255 个字符。 • 只能包含字母数字字符和下划线字符。 • 必须以字母开头。

字段名称	类型	必需	描述
			- 不能以下划线字符结尾。 - 不能以 WPA_ 开头。
description	String	否	字段的描述。最多 1000 个字符。示例： <pre>"description": "The worker's full name."</pre>
type	对象	是	字段的类型。选项包括： - 文本 - 数值 - 布尔值 - 日期 示例： <pre>"type": { "id": "Schema_Field_Type=Da</pre>
parseFormat	String	是 (针对日期类型)	日期的格式。示例： <pre>"parseFormat": "dd-MM-yy HH24:mm:ss"</pre>
precision	数字	是 (针对数值类型)	数值的最大位数。包括小数点左右两侧的所有数字，但不包括小数点本身。最大值为 38。示例： <pre>"precision": 38</pre>
scale	数字	是 (针对数值类型)	数值中小数点右侧的位数。必须小于 precision 值示例： <pre>"scale": 2</pre>
defaultValue	文本	否	如果上传文件中未提供字段值，则此值即为字段的默认值。 使用双引号表示默认文本值。示例：“N/A”。 默认数值必须与指定的数值一致。

字段名称	类型	必需	描述
			precision 与 scale。 默认日期值必须采用以下格式：yyyy-MM-dd'T'HH:mm:ss'Z'。
required	布尔值	否	指定字段是否必须包含值。设置为 True 则设置为必填字段，否则设置为 False。如果某个字段为必填字段，则在任何插入、更新和更新插入的数据操作中，该字段的值都不能为空。
externalID	布尔值	是（利用更新、更新和插入以及删除操作的前提条件）。	指定字段是否表示外部编号。设置为 True 以“外部编号”指定字段，否则设置为 False。 由于“外部编号”字段不得为空，您应指定所有“外部编号”字段为必填字段。您只能将 1 个字段指定为外部编号。 如果您想利用更新、更新和插入以及删除功能，请在新表格中指定外部编号。 要在现有表格中使用更新、更新和插入以及删除功能，您必须对表格进行修改，使其具有一个指定为外部编号的字段。Prism 公开 API 不支持修改包含数据的表格。在这种情况下，请在 Workday 中进行修改。

示例

您的 CSV 源文件描述了以下表格：

员工	员工编号	职务名称	雇用日期	年薪
Logan McNeil	21001	人力资源副总裁	1/1/2010	213298.00
Oliver Reynolds	21003	首席信息官	1/1/2010	394935.00

员工	员工编号	职务名称	雇用日期	年薪
Maximilian Schneider	21004	首席运营官	1/1/2010	255373.00
Teresa Serrano	21005	财务主管	1/1/2010	291785.00

您可以使用 JSON 对其进行描述，如下所示：

```
{
  "parseOptions": {
    "fieldsDelimitedBy": ",",
    "fieldsEnclosedBy": "\"\"",
    "headerLinesToIgnore": 1,
    "charset": {
      "id": "Encoding=UTF-8"
    },
    "type": {
      "id": "Schema_File_Type=Delimited"
    }
  },
  "fields": [
    {
      "ordinal": 1,
      "name": "Employee",
      "description": "Employee",
      "precision": 255,
      "scale": 0,
      "parseFormat": null,
      "type": {
        "id": "Schema_Field_Type=Text"
      }
    },
    {
      "ordinal": 2,
      "name": "Employee_ID",
      "description": "Employee ID",
      "precision": 38,
      "scale": 0,
      "parseFormat": null,
      "type": {
        "id": "Schema_Field_Type=Numeric"
      }
    },
    {
      "ordinal": 3,
      "name": "Job_Title",
      "description": "Job Title",
      "precision": 255,
      "scale": 0,
      "parseFormat": null,
      "type": {
        "id": "Schema_Field_Type=Text"
      }
    },
    {
      "ordinal": 4,
      "name": "Hire_Date",
      "description": "Hire Date",
      "precision": 0,
      "scale": 0,
      "parseFormat": "MM/dd/yy",
      "type": {
```

```
      "id": "Schema_Field_Type=Date"
    },
  ],
  {
    "ordinal": 5,
    "name": "Annual_Salary",
    "description": "Annual Salary",
    "precision": 38,
    "scale": 2,
    "type": {
      "id": "Schema_Field_Type=Numeric"
    }
  }
],
"schemaVersion": {
  "id": "Schema_Version=1.0"
}
}
```

relatedlinks

参考信息

参考信息：表格与数据集中支持的外部数据日期格式

配置具有 **PrismAnalytics** 子程序集的程序集

关于此任务

要将 CSV 数据发送到 Prism Analytics，您的程序集必须包含符合以下条件的“PrismAnalytics”构成：

- 接收 CSV 数据作为输入，所用形式可以是消息的 rootpart，也可以是变量。
- 接收包含 CSV 数据的 JSON 描述的变量。
- 具有多个正确配置的参数。

过程

1. 配置程序集，以获取一个或多个 CSV 源文件。根据您在获取 CSV 数据时所用的方式，您可能需要提供变量名称，也可能需要设置多用途互联网邮件扩展（MIME 类型）。

完成此任务时，请参考以下信息：

CSV 获取方式	注释
“sftp-out”构成	• 在构成的“Input File Pattern”属性中指定 CSV 文件名。您可以使用通配符。 • 使用该组件的 get 方法来检索单个文件或其 mget 方法来检索多个文件。 • Studio 会将 CSV 数据添加到消息根部分。 • 不需要指定变量名称。 • Workday Studio 会自动正确设置 MIME 类型。
检索服务或侦听器服务	• Studio 会将 CSV 数据添加到变量。 • 使用以下内容命名变量 wd.prism.component.csv.input.varname 参数。 • Studio 会自动正确设置 MIME 类型。
其他任何方法	• Studio 会将 CSV 数据添加到消息根或变量。

CSV 获取方式	注释
	- 如果适用，请使用以下内容命名变量 <code>wd.prism.component.csv.input.varname</code> 参数。 - 将根部分或变量的 MIME 类型设置为： <code>text/csv</code> <code>plain/csv</code> <code>application/zip</code>

2. 在“write”步骤的“Message Builder”选项卡中创建文本消息，并添加 JSON 文本来描述数据。输出变量。示例：`temp.schema`。

注：您可以将 JSON 文本直接添加到消息构建器中。也可以在所选工具中编写后再粘贴到 Message Builder。

3. 配置“PrismAnalytics”构成的参数：

参数	描述
<code>wd.prism.component.schema.varname</code>	包含描述数据的 JSON 架构的变量。
<code>wd.prism.component.table.name</code>	Prism 表的名称。表格名称： - 在数据目录中必须唯一。 - 最多可包含 255 个字符。 - 只能包含字母数字字符和下划线字符。 - 必须以字母开头。 - 不能以下划线字符结尾。
<code>wd.prism.component.client.id</code>	Workday 指定给 API 客户端的客户端编号。访问“查看 API 客户端”报告，然后查看“用于集成的 API 客户端”选项卡。
<code>wd.prism.component.client.secret</code>	Workday 指定给 API 客户端的客户端密码。访问“查看 API 客户端”报告，然后查看“用于集成的 API 客户端”选项卡。
<code>wd.prism.component.refresh.token</code>	Workday 指定给 API 客户端的刷新令牌。访问“查看 API 客户端”报告，然后查看“用于集成的 API 客户端”选项卡。
<code>wd.prism.component.operation</code>	对 Prism 表执行的操作。目前仅支持“Replace”选项。此操作可创建或替换同名表格。
<code>wd.prism.component.csv.input.varname</code>	(可选) 包含 CSV 输入的变量。如果要为 CSV 数据添加到消息 <code>rootpart</code>，则不需要包含此参数。 注：当您为“PrismAnalytics”构成添加到程序集时，Workday Studio 不会自动选择包含此参数。请根据需要手动选择，或稍后在组件的“参数”选项卡上添加。
<code>wd.prism.component.api.hostname</code>	Workday REST API 端点访问“查看 API 客户端”报告。移除 <code>https://</code> 位于开头，并且 <code>/ccx/api/v1/<tenant-name></code>” 示例： <code>i-09e55d32d61e2e287.workdaysuv.com</code>。

参数	描述
<code>wd.prism.component.table.description</code>	(可选) 表格描述。最多可包含 255 个字符。
<code>wd.prism.component.table.description.displayName</code>	(可选) 表格的显示名称。最多可包含 255 个字符。
<code>wd.prism.component.table.field.varname</code>	(可选) 定义 Prism 表格。将定义写入变量，然后使用 <code>vars['variable-name']</code> 将其传递给此参数。

下一步做什么

部署并启动集成。

参考信息：**PrismAnalytics** 子程序集属性

“Common”选项卡

属性A	XML 属性名称	描述
编号	<code>id</code>	PrismAnalytics 子装配件的唯一编号。
将回复发送至	<code>routes-response-to</code>	装配件内的子装配件响应消息的目的地。PrismAnalytics 子装配件
执行时间	<code>execute-when</code>	用于确定是否执行 PrismAnalytics 子程序集的条件。
端点	<code>endpoint</code>	子装配件的目标端点地址。PrismAnalytics 组件支持 <code>vm://</code> 端点地址。示例： <code>vm://wcc/PrismAnalytics</code> 。

“Advanced”选项卡

属性A	XML 属性名称	描述
存储消息	<code>store-message</code>	除非 Workday 技术支持另行通知，否则 无 指定内容。 要存储消息，请使用 Workday Studio 的“store”步骤。
传输类	<code>transport</code>	如果需要，为每个装配件指定一个备选的实现类。 您可以通过编辑 <code>WEB-INF/classes/spring/assembly-bean.xml</code> 文件来全局更改实现类。
忽略动态端点	<code>ignore-dynamic-endpoints</code>	指定是忽略传入消息标头中的动态 WS-Addressing (WSA) 值，还是使用它来重设传输协议的 Endpoint 属性中的值。

属性A	XML 属性名称	描述
		使用“set-dynamic-endpoint”步骤指定 WS-Addressing 详细信息。
复制申请	clone-request	指定是否在子装配件之间进行链式处理。设置为“true”可创建原始消息的副本，这样任何修改请求消息的子程序集都不会影响其他子程序集。设置为“false”可传递原始消息。
传播中止	propagate-abort	指定中止标志是否从子程序集中传播到调用程序集。 子装配件在标记 MediationContext 状态为“已中止”。
取消设置属性	unset-properties	指定在 本地转出 传输函数调用子部件后，是否重置所有的 本地输出 属性。

“Parameters”选项卡

参数	描述
wd.prism.component.schema.varname	包含描述数据的 JSON 架构的变量。
wd.prism.component.table.name	Prism 表在数据目录中的标识符。表格名称： - 在数据目录中必须唯一。 - 最多可包含 255 个字符。 - 只能包含字母数字字符和下划线字符。 - 必须以字母开头。 - 不能以下划线字符结尾。 此组件不能用于更新 Prism 表名称，而只能用于创建 Prism 表名称。
wd.prism.component.client.id	Workday 指定给 API 客户端的客户端编号。访问“查看 API 客户端”报告，然后查看“用于集成的 API 客户端”选项卡。
wd.prism.component.client.secret	Workday 指定给 API 客户端的客户端密码。
wd.prism.component.refresh.token	Workday 指定给 API 客户端的刷新令牌。访问“查看 API 客户端”报告，然后查看“用于集成的 API 客户端”选项卡。
wd.prism.component.operation	对 Prism 数据集执行的操作。示例：Replace，Append。 Replace 创建表格或替换具有相同名称的表格。
wd.prism.component.csv.input.varname	(可选) 包含 CSV 输入的变量。如果要添加 CSV 数据到消息 rootpart，则不需要包含此参数。 注：当您将“PrismAnalytics”构成添加到程序集时，Workday Studio 不会自动选择包含此参数。请

参数	描述
	根据需要手动选择，或稍后在组件的“参数”选项卡上添加。
<code>wd.prism.component.api.hostname</code>	Workday REST API 端点访问“查看 API 客户端”报告。移除 <code>https://</code> 位于开头，并且 <code>/ccx/api/v1/<tenant-name></code> “” 示例： <code>i-09e55d32d61e2e287.workdaysuv.com。</code>
<code>wd.prism.component.table.description</code>	(可选) 表格描述。最多可包含 255 个字符。
<code>wd.prism.component.table.display.name</code>	(可选) 表格的显示名称。最多可包含 255 个字符。 您不能使用此组件来更新 Prism 表的显示名称，而只能用于创建名称。
<code>wd.prism.component.table.field.varname</code>	(可选) 定义 Prism 表格。将定义写入变量，然后使用 <code>vars[variable-name]</code> 将其传递给此参数。

错误处理程序构成

概念：错误处理程序构成

Workday Studio 在“Palette”中提供了 3 种不同的错误处理程序：

名称	函数
<code>send-error</code>	让您能够使用标准构成和步骤来处理错误。一种常见的模式是通过调用“PutIntegrationMessage”通用构成，使用“send-error”向集成事件报告错误。
<code>log-error</code>	可将当前错误记录到输出日志中。包括错误消息和 Java 异常的堆栈跟踪。如果无需逐个显示集成事件中的错误，但您想在错误发生时记录消息，请使用此处理程序。避免记录大型消息，因为它们会降低性能，而且可能会被截断，从而导致诊断数据丢失。
<code>custom-error-handler</code>	让您能够提供使用 Java 来处理错误的 Spring Bean。如果处理错误时需要复杂的 Java 代码，请使用此处理程序。

您可以将错误处理程序添加为中介构成的子元素，或程序集中的单个元素。添加到中介构成的错误处理程序位于本地。程序集中的错误处理程序是全局性的。

当 Workday 在 Workday Studio 集成中遇到错误时，它会停止正常处理操作并展开已处理的构成链，以便查找要调用的错误处理程序。Workday 始终会先尝试在本地处理错误。如果没有可用的本地错误处理程序，则交由全局错误处理程序来处理。如果您希望精细控制发生错误时的处理方式，或者需要报告集成在出错时所执行操作的详细相关信息，请使用本地错误处理程序。可将全局错误处理程序视为故障保护。

如果所有错误处理程序都已触发，但仍有错误未被标记为已处理，则 Workday 会处理相关消息，根据异常生成错误，并将程序集标记为已终止。它会将集成描述为“已完成但有错误”。

需要注意的默认行为：

- 本地错误处理程序仅处理其父级中介构成中出现的错误。但是，它们也可以处理下游构成中的错误。要启用此行为，请将包含错误处理程序的中介构成的“Handle Downflow Errors”属性设置为“true”。

- 当您调用错误处理程序时，Workday 会将错误标记为“已处理”。但是，如果您将错误处理程序的“Rethrow Error”属性设置为“true”，则 Workday 不会清除该错误，因此，该错误将由下一个范围内上游错误处理程序或全局错误处理程序进行处理。使用此模式可在本地报告错误的详细信息，但可在程序集中的较高级别处理整体失败或完成情况。
- 当 Workday 处理错误时，它会从调用的错误处理程序重新启动消息处理，继续沿着响应路径找到出错的元素。但是，您可以指定从引发错误的元素处继续进行处理。要启用此行为，请将包含错误处理程序的中介构成的“Continue After Error”属性设置为“recover”。如果错误处理程序调用的代码可以更正错误条件，从而能够成功重试失败的操作，请使用此行为。

您可以通过在“Properties”视图中添加 MVEL 条件表达式，来进一步控制错误处理程序。当存在条件表达式时，仅当它们的计算结果都为 true 时，Workday 才会调用错误处理程序。

您可以使用强制引发错误 `context.setError` 或 `context.SetException` MVEL 或 Java 代码中的方法。当使用 `setError` 方法，您可以配置一个可由错误处理程序中的条件表达式检测到的错误编号。如果您在本地处理错误，但希望将错误类型的某些详细信息传递给上游错误处理程序进行常规处理，则可以使用此技巧。

您还可以使用“Route”构成来处理错误：

- “failover-strategy”会尝试使用路由，如果发生错误，则会执行故障转移路由，并将错误标记为“已处理”。
- 通过 `custom-strategy`，您可以提供一个名为“”的 `java - java - java RoutingStrategy` 接口，其具有 `isHandleError` 方法。使用此方法可指定该策略处理错误。

参考信息：**Custom-Error-Handler** 构成属性

“Common”选项卡

属性A	XML 属性名称	描述
错误处理程序编号	id	程序集内的 custom-error-handler 组件的唯一编号。
重新引发错误	rethrow-error	指定 custom-error-handler 组件在完成处理后是否将错误传递回任何上游错误处理程序。默认值为 false，这表示 custom-error-handler 组件会单独处理错误，然后将错误标记为已处理。 您还可以通过调用 <code>setErrorHandled</code> 的方法 <code>MediationContext</code> 对象。示例： <code>mediationContext.setErrorHandled(true)</code>
java - java - java	ref	实现以下 2 个接口类之一的 Spingle 的 ID： • <code>OnErrorHandler</code>. • <code>ConditionalOnErrorHandler</code>，它的扩展名为 <code>OnErrorHandler</code> 使用布尔值方法 <code>invokeHandleError(MediationContext)</code>。如果此方法返回 true，则 Workday 会调用

属性A	XML 属性名称	描述
		handleError(MediationContext context) 方法。 注: 如果您实施 ConditionalOnErrorHandler 界面，以便重设您定义的任何条件表达式。

“表达式”选项卡

属性	XML 属性名称	描述
条件表达式	condition-expression	让您能够直接输入 MVEL 条件表达式。 如果所有表达式的计算结果均为 true，如果您不提供表达式，则系统会调用错误处理程序。否则，Workday 不会调用错误处理程序，而是沿着处理链继续进行错误处理，同时 Workday 会添加一个“”类型的 审核日志 条目。ERROR_HANDLER_SKIPPED。 要在任何现有行下方添加新表达式，请单击“添加”，然后输入表达式。 要在现有行上方添加新表达式，请将光标放在该行中，然后单击“Add”下拉菜单中的“Insert”。 使用“Expressions”选项卡右侧的按钮可删除行或对行重新排序。

参考信息：Log-Error 构成属性

“Common”选项卡

属性A	XML 属性名称	描述
错误处理程序编号	id	程序集中的 日志错误 构成编号。
重新引发错误	rethrow-error	指定 Log-error 组件在完成处理后是否将错误传递回任何上游错误处理程序。默认值为 false，表示 Log-error 组件单独处理错误，然后将错误标记为已处理。 您还可以通过调用 setErrorHandled 的方法 MediationContext

属性A	XML 属性名称	描述
		对象。示例： <code>mediationContext.setErrorHandled(tr</code>
级别	level	Log4J 日志记录级别。有以下 5 个选项： • debug • info • warn • error • fatal

“Advanced”选项卡

属性A	XML 属性名称	描述
种类	category	指定 Log4J 记录器名称的一部分。默认值为 DEFAULT。

“表达式”选项卡

属性	XML 属性名称	描述
条件表达式	condition-expression	让您能够直接输入 MVEL 条件表达式。 如果所有表达式的计算结果均为 true，如果您不提供表达式，则系统会调用错误处理程序。否则，Workday 不会调用错误处理程序，而是沿着处理链继续进行错误处理，同时 Workday 会添加一个“”类型的 审核日志 条目。ERROR_HANDLER_SKIPPED。 要在任何现有行下方添加新表达式，请单击“添加”，然后输入表达式。 要在现有行上方添加新表达式，请将光标放在该行中，然后单击“Add”下拉菜单中的“Insert”。 使用“Expressions”选项卡右侧的按钮可删除行或对行重新排序。

参考信息：Send-Error 构成属性

“Common”选项卡

属性A	XML 属性名称	描述
错误处理程序编号	id	程序集中的 send-error 组件的唯一编号。

属性A	XML 属性名称	描述
重新引发错误	rethrow-error	指定 send-error 组件在完成处理后是否将错误传递回任何上游错误处理程序。默认值为 false，这表示 send-error 组件会单独处理错误，然后将错误标记为已处理。 您还可以通过调用 setErrorHandled 的方法 MediationContext 对象。MVEL 示例： context.setErrorHandled(true)。
寄送目的地	routes-to	send-error 组件将消息定向到的组合元素编号。

“Expressions”选项卡

属性	XML 属性名称	描述
条件表达式	condition-expression	让您能够直接输入 MVEL 条件表达式。 如果所有表达式的计算结果均为 true，如果您不提供表达式，则系统会调用错误处理程序。否则，Workday 不会调用错误处理程序，而是沿着处理链继续进行错误处理，同时 Workday 会添加一个“”类型的 审核日志 条目。ERROR_HANDLER_SKIPPED。 要在任何现有行下方添加新表达式，请单击“添加”，然后输入表达式。 要在现有行上方添加新表达式，请将光标放在该行中，然后单击“Add”下拉菜单中的“Insert”。 使用“Expressions”选项卡右侧的按钮可删除行或对行重新排序。

转换步骤

使用文本架构文件将文本转换为 XML

开始之前

创建一个包含“textschema”步骤的程序集。获取将构成文本架构基础的示例文本。

关于此任务

“textschema”步骤让您能够使用文本架构将文本转换为 XML。要开发文本架构，您必须知道要转换的数据的结构。如果有一个包含代表性数据的小型示例文档，则可大大简化这个过程。本主题假设您有这样一个示例文档。

您可以使用“textschema”步骤将 XML 转换为文本，或者转换带分隔符的文件，但在大多数情况下，分别使用“XTT”步骤和“csv-to-xml”步骤会更简单。

过程

1. 在“textschema”步骤的“Properties”视图中，单击“Create Text Schema File”。
2. 选择文本架构文件的父级文件夹并提供文件名。如果需要，指定目标命名空间 URI。
3. Studio 打开文本模式编辑器，并在 树状 视图中显示新的文本模式。
4. 将示例文本剪切并粘贴到文本架构编辑器右上方的区域中，将 initial-text。

注：要从输入文件中去除终止换行符，请将此属性添加到根架构元素：ts:eofStrip="
"。

5. 开始描述示例文本。选择第一个字段，右键单击并选择“Create Field”。

注：如果您没有要转换的文本示例，但知道其结构，则可以直接编辑文本架构的“Source”视图。

6. 在“Create Field Wizard”的“Identity”页面上，提供字段的基本描述。完成此任务时，请参考以下信息：

选项	描述
名称	字段的名称。
Occurrence	选项包括： • Default：字段为必填字段，且仅出现一次。 • Required：与默认选项相同。 • Optional：字段为可选字段，且只能出现一次。 • Multiple Optional：字段为可选字段，且可以出现多次。 • Multiple Required：字段为必填字段，且可以出现多次。
Start Tag	一个正则表达式，其指定的文本会导致生成元素。请注意，与此表达式匹配的文本不是输出值的一部分。如果您希望在输出值中显示匹配的文本，请使用 ts:format 选项。
结束标记	一个正则表达式，其指定的文本用于标记元素的结束。请注意，与此表达式匹配的文本不是输出值的一部分。如果您希望在输出值中显示匹配的文本，请使用 ts:format 选项。
截断	指定是截断固定长度字段中的溢出数据，还是引发错误。默认值为 true。
分隔符	一个正则表达式，其指定的字符用于分隔重复字段。仅当您选择“Multiple Optional”或“Multiple Required”出现次数选项后，此选项才会启用。
Omit	指定在“text-to-XML”或“XML-to-text”转换的输出中是否忽略字段。默认为不忽略。

7. 在“Create Field Wizard”的“Delimit”页面上，描述示例文本中用于表示一个字段的字符数：

选项	描述
格式	一个正则表达式，用于标识字段的有效文本模式。此表达式可用于替换“Start Tag”和“End Tag”，以确保在输出中显示所有匹配的文本。在处理多次出现的值时，您也可以使用此表达式，从而清楚知道序列何时结束。
Fixed Length	指定固定长度字段的长度。引号和转义字符包括在内。
Quoted	指示是否将字符串值视为带引号的字符串。选项包括： • Default： • always： • always-including-empty： • optional： • never：
Padding	一个正则表达式，用于标识固定长度字段内的任何填充文本。
Preferred Padding	指定在“Padding”表达式可匹配多个可能的字符串时要使用的值。
Align	指定需要填充的固定长度字段的对齐方式。

8. 在“Create Field Wizard”的“Interpret”页面上，描述示例文本中用于表示一个字段的字符数：

选项	描述
类型	字段的数据类型。
默认值	字段的默认值。
Date Format	指定日期或日期时间模式。支持 Java 允许的任何模式 SimpleDateFormat。
日期语言	指定在解释日期时使用的日期语言。确保当日期格式使用月份名称时，名称以相应的语言处理。默认值为 EN。
两位数年份	指定用于解释 2 位数年份日期的值。文本模式解析器会将早于该值的 2 位数年份日期解释为在当前世纪中。示例：如果您输入 18，则解析器会解释为 14 作为 2014 与 96 作为 1996。
数字格式	指定数字模式。支持 Java 允许的任何模式 DecimalFormat。
小数分隔符	指定用于十进制数字的分隔符。
Grouping Separator	指定数字的千位分组分隔符。

9. 对示例文本中的每个字段重复第 6 步到第 9 步，从而在树状视图中添加新元素。创建所有必填字段定义后，删除剩余的。Unparsed 子元素。
10. 要将您创建的所有元素封装到一个复杂类型中，请选择这些元素，右键单击它们，然后选择“元素封装”。然后，为包装器元素提供名称和出现次数值。
11. 在包装器元素的“属性”视图中，从“结束标记”下拉列表中选择“/n”。
12. 默认架构根元素的名称为 File。在元素“属性”视图的“架构”选项卡中，提供一个更有意义的名称。

13. 在最高层级的“属性”视图中 schema 元素，则“根元素”属性的值为 tns:File。替换 File 将其替换为您为根元素选择的有意义的名称。

14. 如要从生成的输出中移除任何多余空格，请选择最高层级的“属性”视图。schema 元素。从“填充位数”下拉列表中，选择“+”。保存架构。

概念：转换步骤

通过转换步骤，您可以执行各项操作，在消息通过您的集成时对其进行重构。Workday Studio 在“Palette”中提供以下转换步骤：

转换步骤	函数	注释
xslt	使用 XSL 转换 (XSLT) 样式表转换 XML 文档。	- 支持 XSLT 2.0。 - 兼容版本号不超过 2017.05 的程序集。
XSLT+	使用 XSL 转换 (XSLT) 样式表转换 XML 文档。	- 支持 XSLT 3.0 流式传输功能。 - 兼容版本号晚于 2017.05 的程序集。
fop	基于 XML 文件生成 PDF 和 RTF 文档。	- 调用 Apache 格式化对象处理器 (FOP) 实现。 - 实现 Apache FOP XML 配置文件，用以注册自定义字体。
text-excel	将 Excel 文件转换为逗号分隔值 (CSV) 文本格式文件。还可将 CSV 文件转换为 Excel 文件。	
textschemata	使用文本架构文件，将文本转换为 XML 或将 XML 转换为文本。	
wrap-soap	将 XML 消息封装在 SOAP 信封中，以转发到基于 SOAP 的 Web 服务。	
xml-to-java	使用 JAXB 将 XML 解封送为 Java 对象。	- 实现 XPath，以从消息中提取 XML。 - 实施 JAXB，以将 XML 转换为 Java 对象。 - 将转换后的 Java 对象插入 MediationContext。
json-transformer	将 JSON 文档作为输入，并通过将值映射到 JSON 路径来对其进行转换。	
xml-to-json	将 XML 文档转换为 JSON 格式。	
json-to-xml	将 JSON 文档转换为 XML 格式。	
java-to-xml	使用 JAXB 将 Java 对象封送至 XML。	- 从中提取 Java 对象 MediationContext。 - 实施 JAXB，以将 Java 对象转换为 XML 对象。

转换步骤	函数	注释
		实现 XPath，以将转换后的 XML 插入到消息中。
xml-to-csv	将 XML 文档转换为 CSV 格式，同时流式传输输入和输出文档，从而减少内存占用。	
csv-to-xml	将 CSV 文档转换为 XML 格式，同时流式传输输入和输出文档，从而减少内存占用。	
character-conversion	对传入消息的文本内容执行字符转换操作。	
mtable-builder	创建并存储 mtable 中的对象 MediationContext 属性。	- 支持以下子级元素： - csv-mtable-reader - json-mtable-reader
mtable-writer	检索 mtable 中的对象 MediationContext 属性。 显示 a 的内容 mtable 输出消息中的对象。	- 支持以下子级元素： - csv-mtable-writer - json-mtable-writer
etv	根据 XML 文档中的元素所附加的属性，转换和验证这些元素。	
xtr	根据 XML 中定义的属性，将 XML 转换为文本。	

参考信息：XSLT 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	xslt 中介步骤在组合件中的唯一编号。
输入	input	指定 xslt 中介步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。

属性A	XML 属性名称	描述
		变量：使用中的已命名变量 <code>MediationContext</code>。
输出	output	指定 xslt 中介步骤将输出定向到何处。共有 5 个可能的值： <code>message</code>：整条消息，包括 <code>rootpart</code> 和附件。此值会创建一条新消息，同时会移除所有附件。 <code>soapbody</code>：中 SOAP 正文的内容 <code>rootpart</code> 消息的“”。 <code>attachment</code>：消息的 MIME 附件。您必须指定附件索引。 <code>rootpart</code>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 <code>MediationContext</code>。
URL	url	xslt 中介步骤使用的 XSLT 文件的位置。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 xslt 中介步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	确定是否执行 XSLT 步骤的条件。
使用文件支持的托管数据	use-file-backed-managed-data	指定在达到数据阈值后是否将所有输出流式传输到本地磁盘。
消息属性	messages-property	该 <code>MediationContext</code> 属性（您要在配置）List 对象。
重用策略	reuse-strategy	指定 XSLT 转换器重用策略。有如下 4 个选项： - 默认值：<code>MediationContext</code> 会自动确定您的重用策略 <code>no-reuse</code>：您的服务器不会重复使用 Transformer 对象。

属性A	XML 属性名称	描述
		• Templates：您的服务器会构建 Templates 对象，从中创建新的转换器对象。 • Transformer- Pool：您的服务器使用一个 Transformer 池。仅当转换器对象不缓存值时，我们才建议您使用此选项。 示例：Saxon 缓存以下操作的结果：<code>document()</code> 调用。
安全处理	<code>secure-processing</code>	指定是否启用可能会影响安全性的处理。 如果设置为“true”，“xslt”中介步骤会安全地处理 XML，并对 XML 构造设置限制，以避免出现拒绝服务攻击等状况。 如果设置为“false”，“xslt”中介步骤会忽略安全问题，例如对 XML 构造的限制。
Workday Transformer	<code>transformer-factory</code>	XSLT 的类名称 TransformerFactory（由 xslt 中介步骤使用） 示例：Saxon 的值为 <code>net.sf.saxon.TransformerFactoryImpl</code>

参考信息：XSLT+ 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	<code>id</code>	XSLT+ 中介步骤在程序集中的唯一编号。
输入	<code>input</code>	指定 XSLT+ 中介步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 rootpart 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展

属性A	XML 属性名称	描述
		(MIME) 类型的邮件，则系统会选中整封邮件。 变量：使用中的已命名变量 <code>MediationContext</code>。
输出	<code>output</code>	指定 XSLT+ 中介步骤将输出定向到何处。共有 5 个可能的值： <code>message</code>：整条消息，包括 <code>rootpart</code> 和附件。此值会创建一条新消息，同时会移除所有附件。 <code>soapbody</code>：中 SOAP 正文的内容 <code>rootpart</code> 消息的“”。 <code>attachment</code>：消息的 MIME 附件。您必须指定附件索引。 <code>rootpart</code>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 <code>MediationContext</code>。
URL	<code>url</code>	XSLT+ 中介步骤使用的 XSLT 文件的位置。
安全处理	<code>secure-processing</code>	指定是否启用可能会影响安全性的处理。 如果设置为“true”，“XSLT+”中介步骤会安全地处理 XML，并对 XML 构造设置限制，以避免出现拒绝服务攻击等状况。 如果设置为“false”，“XSLT+”中介步骤会忽略安全问题，例如对 XML 构造的限制。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	<code>description</code>	对合并日志中显示的 XSLT+ 中介步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	<code>execute-when</code>	确定是否执行 XSLT+ 步骤的条件。
消息属性	<code>messages-property</code>	该 <code>MediationContext</code> 将要配置的属性 List 对象。

属性A	XML 属性名称	描述
架构验证模式	schema-validation-mode	指定是否根据提供的架构使用 Saxon XSLT 处理器来验证输入和输出。

“Schema URLs”选项卡

属性	XML 属性名称	描述
模式 URL	schema-url	XSLT+ 中介步骤使用架构文件的位置来验证输出文档。

参考信息：FOP 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中的 中介 步骤的唯一编号。
输入	input	指定 fop 中介步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。 <i>变量</i>：使用中 的已命名变量 MediationContext。
输出	output	指定 中介 步骤将输出定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。 <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。

属性A	XML 属性名称	描述
		- rootpart : MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 MediationContext。
创建者	author	指定 author 文档元数据中的值。
基本 URL	base-url	用于解析相对 URL 的基本 URL。如果不存在，此位置还会用于相对字体位置。font-base-url 已指定。可以是项目的相对路径 WSAR-INF 或绝对 URL。示例： ./fop
创建日期	creation-date	指定 creation date 文档元数据中的值。输入使用美国日期格式的值。示例：5/14/08 4:50 PM。
创建人	creator	指定 creator 文档元数据中的值。
关键字	keywords	指定 keywords 文档元数据中的值。
标题	title	指定 title 文档元数据中的值。
用户密码	user-password	使您能够为 PDF 和 RTF 文档设置用户密码。该密码可启用以下文档权限： - allow-copy-content - allow-edit-annotations - allow-edit-content - allow-print

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对 fop 中介步骤显示在合并日志中的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	确定是否执行 fop 步骤的条件。
允许复制内容	allow-copy-content	指定当您使用 user-password。

属性A	XML 属性名称	描述
允许编辑注释	allow-edit-annotations	指定当您使用 user-password。
允许编辑内容	allow-edit-content	指定您在使用 user-password。
允许打印	allow-print	指定在使用 user-password。
配置 Xml	configuration-xml	配置文件的 XML 配置文件的路径，该路径必须位于本地磁盘上，而不是 URL。此配置文件是注册自定义字体的唯一方法。
创建日期格式	creation-date-format	用于解析的日期格式 creation-date XML 属性支持以下值： • full • long • medium • short
创建日期的区域设置 - 国家/地区	creation-date-locale-country	该 country 用于解析的值 creation-date XML 属性
创建日期的区域设置 - 语言	creation-date-locale-language	该 language 用于解析的值 creation-date XML 属性此值必须是由两个字母的小写字母组成的 ISO-639 语言代码。
创建时间格式	creation-time-format	用于解析的时间格式 creation-date XML 属性支持以下值： • full • long • medium • short
字体基本 URL	font-base-url	用于解析相对字体 URL 的基本 URL。示例： ./fop/fonts
连字符基本 URL	hyphenation-base-url	基本 URL，用于将相对 URL 解析为连字符模式文件。
所有者密码	owner-password	使您能够为 PDF 和 RTF 文档设置所有者密码。该密码可启用以下文档权限： • allow-copy-content • allow-edit-annotations • allow-edit-content • allow-print
生产者	producer	指定 producer 文档元数据中的值。

属性A	XML 属性名称	描述
目标解决方法	target-resolution	由位图呈现器 (例如 TIFF 呈现器和 AAPACH BAPI) 生成的位图图片的输出分辨率, 以每英寸点数 (DPI) 为单位。

参考信息：Text-Excel 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中的 text-Excel 中介步骤的唯一编号。
输入	input	指定 text-Excel 中介步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 rootpart 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。 • <i>变量</i>：使用中 的已命名变量 MediationContext。
输出	output	指定 文本 Excel 中介步骤的定向输出。共有 5 个可能的值： • <i>message</i>：整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 • <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。 • <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。

属性A	XML 属性名称	描述
		变量：将输出另存为中的已命名变量 MediationContext。
样式	style	文本-Excel 中介步骤的转换样式。有以下 3 个选项： - XLSX/XLS to Text：将 XLSX 或 XLS 格式的 Excel 文件转换成 CSV 文件。 - Text to XLS：将 CSV 文件转换成 XLS 格式的 Excel 文件。 - Text to XLSX：将 CSV 文件转换成 XLSX 格式的 Excel 文件。 注：在 XLSX 格式与其他格式之间来回转换时，此构成会流式传输。
Excel 样式	excel-style	指定 文本 Excel 中介步骤要转换整个 Excel 工作簿还是单个 Excel 工作表。
Excel 工作表编号	excel-sheet-number	Excel 文件中要转换的目标工作表编号。 如果您将“Excel Style”属性配置为“workbook”，Workday 将忽略“text-Excel”中介步骤的“Excel Sheet Number”设置。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	在合并的日志中显示的 文本格式步骤描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	确定是否执行 text-Excel 步骤的条件。
使用值	usevalues	指定 文本 Excel 中介步骤是否使用电子表格的基础值（而不是屏幕上显示的格式化值）。

relatedlinks

参考信息

[Workday 31 What's New Post: Workday Studio](#)

参考信息：TextSchema 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	textschemata 中介步骤在程序集中的唯一编号。
输入	input	指定 textschemata 中介步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。 • 变量：使用中 的已命名变量 MediationContext。
输出	output	指定 textschemata 中介步骤将输出定向到何处。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 • 变量：将输出另存为中的已命名变量 MediationContext。
URL	url	textschemata 中介步骤使用的文本架构 XSD 文件的位置，以相对路径或绝对 URL 的形式表

属性A	XML 属性名称	描述
		示。您可以使用 MVEL 表达式从 MediationContext。示例： <code>@{parts[1].xpath('/script/location')}</code>
样式	style	textschem 中介步骤的转换样式。您有以下 2 个选择： - text2xml：将文本转换为 XML。 - xml2text：将 XML 转换为文本。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 textschem 中介步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	一个用于确定 文本架构 步骤是否执行的条件。
使用文件支持的托管数据	use-file-backed-managed-data	指定在达到数据阈值后是否将所有输出流式传输到本地磁盘。

参考信息：Wrap-Soap 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集内的 Workday-soap 中介步骤的唯一编号。
输入	input	指定 Workday-Soap 中介步骤从何处获取数据作为输入。共有 5 个可能的值： - message：整条消息，包括 rootpart 和附件。 - soapbody：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 - attachment：消息中的 MIME 附件。您必须指定附件索引。 - rootpart：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。

属性A	XML 属性名称	描述
		变量：使用中 的已命名变量 MediationContext。
输出	output	指定 Workday-Soap 中介步骤将输出定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。 <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 MediationContext。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对显示在合并日志中的 Workday-Soap 中介步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	此条件决定是否执行 Workday-soap 步骤。
预算	interface	SOAP正文元素的直接子级
简单概念操作	soap-action	SOAP 操作标头的值。您必须为 SOAP HTTP 请求指定此值。
Soap 版本	soap-version	SOAP 信封的版本。

参考信息：XML-to-Java 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中 xml 到 java 中介步骤的唯一编号。

属性A	XML 属性名称	描述
输入	input	指定 xml-to java 中介步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 rootpart 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。 • <i>变量</i>：使用中 的已命名变量 MediationContext。
属性	property	的名称 MediationContext 属性，供 xml-to java 中介步骤插入 Java 对象。
薪酬组合	packages	用分号分隔的 xml-to java 中介步骤所需的 Java 包列表，用于使用 JAXB 将 XML 转换为 Java 对象。
展开 JAXB 元素	unwrap-jaxbelement	指定“xml 到 java”中介步骤是否将 Java 对象存储在 JAXB 元素中。
命名空间	namespaces	XPath 表达式使用的命名空间。
XPath	xpath	个 XPath 表达式，用于从消息中提取 XML，并将其转换为 Java 对象。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的“xml 到 java”中介步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	该条件决定是否执行“xml 到 java”步骤。
XPath 版本	xpath-version	“xml-to-java”中介步骤使用的 XPath 版本。示例：“Version 1.0”“Version 2.0”或“Version 3.0”。

参考信息：JSON 转换器属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	json-transformer 中介步骤在程序集中的唯一编号。
输入	input	指定 json-transformer 中介步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。 • 变量：使用中 的已命名变量 MediationContext。
输出	output	指定 json-transformer 中介步骤将输出定向到何处。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 • 变量：将输出另存为中的已命名变量 MediationContext。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 json-transformer 中介步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	确定是否执行 json-transformer 步骤的条件。

“Mappings”选项卡

属性	XML 属性名称	描述
Json 路径	json-path	指定 JSON 树中将指定关联 值的位置。
无匹配项失败	fail-no-match	确定当指定的 JSON 路径未返回节点时发生的情况： • true：引发错误。 • fail：忽略缺失的路径。

参考信息：XML-to-JSON 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中的“xml-to-json”中介步骤的唯一编号。
输入	input	指定“xml-to JSON”中介步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 rootpart 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。 • <i>变量</i>：使用中 的已命名变量 MediationContext。

属性A	XML 属性名称	描述
输出	output	指定“xml-to JSON”中介步骤将输出定向到何处。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 • 变量：将输出另存为中的已命名变量 <i>MediationContext</i>。
模式 URL	schema-url	指定相对 URL，以便“xml-to JSON”中介步骤将 XML 架构文件作为程序集 Web 服务项目中的资源进行检索。
模式	schema	指定“xml 转 JSON”步骤从何处获取 XML 架构。以下是 5 个可能的值： • <i>message</i> • <i>soapbody</i> • <i>attachment</i> • <i>rootpart</i> • <i>variable</i>
忽略属性	ignore-attributes	指定在将 XML 文档转换为 JSON 时，“xml 到 JSON”步骤是否忽略属性。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的“xml-to-json”中介步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	此条件决定是否执行“xml 到 json”步骤。

参考信息：JSON-to-XML 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	json-to-xml 中介步骤在程序集中的唯一编号。
输入	input	指定 “json-to-xml” 中介步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。 • 变量：使用中 的已命名变量 MediationContext。
输出	output	指定 json-to-xml 中介步骤将输出定向到何处。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 • 变量：将输出另存为中的已命名变量 MediationContext。
根元素名称	root-element-name	XML 文档最外层换行标记的名称。

属性A	XML 属性名称	描述
嵌套数组名称	nested-array-name	JSON 输入中匿名嵌套数组的名称。
嵌套对象名称	nested-object-name	JSON 输入中匿名嵌套对象的名称。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 “json-to-xml” 中介步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	该条件决定是否执行 “json-to-xml” 步骤。

参考信息：Java-to-XML 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	java 到 xml 中介步骤在程序集中的唯一编号。
输出	output	指定 java-to-xml 中介步骤将输出定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。 <i>attachment</i>：消息的 MIME 附件。指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 <i>变量</i>：将输出另存为中的已命名变量 MediationContext。
属性	property	的名称 MediationContext 属性（java 到 xml 中介步骤从中提取 Java 对象）。
薪酬组合	packages	用分号分隔的 Java 包列表，这些包是 java-to xml 中介步骤（使

属性A	XML 属性名称	描述
		用 JAXB) 将 Java 对象转换为 XML 时所需的。
根元素名称	root-element-name	根元素的名称，该元素用于将 Java 对象封送至 XML。
根元素命名空间	root-element-namespace	根元素的命名空间，该命名空间用于将 Java 对象封送至 XML。
命名空间	namespaces	XPath 表达式使用的命名空间。
XPath	xpath	用于将 XML 插入到消息中的 XPath 表达式。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的“java 到 xml”中介步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	该条件决定是否执行“java-to-xml”步骤。
XPath 版本	xpath-version	指定使用 XPath 1.0、2.0 还是 3.0 版本。对于 Workday 11 及之前的程序集版本，请输入值 1。对于 Workday 12 及之后的程序集版本，请输入值 2 或 3。 您可以允许 Workday 使用可插入系统属性值的 MVEL 模板动态设置版本。示例： <code>@{props.get('the_version')}</code> 。

参考信息：XML-to-CSV 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中“xml 到 csv”步骤的唯一编号。
输入	input	指定“xml 转 csv”步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。

属性A	XML 属性名称	描述
		• <i>attachment</i> : 消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i> : MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件, 则系统会选中整封邮件。 • 变量 : 使用中 的已命名变量 MediationContext。
输出	output	指定 “xml 转 csv” 步骤将文件定向到何处。共有 5 个可能的值 : • <i>message</i> : 整条消息, 包括 <i>rootpart</i> 和附件。此值会创建一条新消息, 同时会移除所有附件。 • <i>soapbody</i> : 中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i> : 消息的 MIME 附件。指定附件索引。 • <i>rootpart</i> : MIME 消息根部分的内容, 如果消息不是 MIME, 则为消息本身。此值会保留原始消息中的所有附件。 • 变量 : 将输出另存为中的已命名变量 MediationContext。
分隔符	separator	要在 CSV 输出文件中每个数据条目之后插入的分隔符。
行分隔符	line-separator	要在 CSV 输出文件中的每个数据条目之后插入的新行分隔符。支持以下值 : • LF 适用于 Windows、Mac OS X 及更高版本。 • CR (Mac OS 版本 9 或更低版本) • CRLF 对于 Microsoft Windows。
写入标头行	writeHeaderLine	指定 “xml 到 csv” 步骤是否将 XML 文档中连续子级元素的名称作为逗号分隔值插入到输出文档的第一行中。
格式	format	指定您的 CSV 输出文档是否符合 RFC-4180 要求。您可以输入以下任一值 :

属性A	XML 属性名称	描述
		• rfc4180 • simple 如果您输入“rfc4180”，则 CSV 输出文档将符合 RFC-4180 标准。 如果您输入“simple”或将该字段留空，则“xml-to-csv”步骤将执行以下操作： • 将撇号视为备用引号。 • 去除值开头的空格。 • 接受任何行结束符 (CR、LF 或 CRLF)。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的“xml 到 csv”步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	此条件决定是否执行“xml 到 csv”步骤。

参考信息：CSV-to-XML 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中“csv-to-xml”步骤的唯一编号。
输入	input	指定“csv-to-xml”步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 rootpart 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。

属性A	XML 属性名称	描述
		变量：使用中的已命名变量 MediationContext。
输出	output	指定“csv 转 XML”步骤将文件定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。 <i>attachment</i>：消息的 MIME 附件。指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 MediationContext。
列名称	colNames	XML 输出文件中显示的列的名称。 请以英文逗号分隔的名称列表形式为此属性输入值。示例： col1, col2, col3.
命名空间前缀	namespacePrefix	在 XML 输出文件中显示的命名空间前缀。
命名空间 URI	namespaceURI	显示在 XML 输出文件的根元素上的命名空间 URI。 示例：如果您将 URI 指定为 http://example.workday.com/SupplyChainManagement/DataTypes.xsd 命名空间前缀 ns，则“csv-to-xml”步骤会在您的 XML 输出文件中创建以下根元素： <pre><ns:root xmlns:ns="http:// example.workday.com/ SupplyChainManagement/ DataTypes.xsd"></pre>
根名称	rootName	在 XML 输出文件中显示的根元素的名称。
行名称	rowName	XML 输出文件中显示的行名称。

属性A	XML 属性名称	描述
		请以英文逗号分隔的名称列表形式为此属性输入值。示例： row1, row2, row3.
分隔符	separator	在 CSV 输入文件中使用的分隔符。
格式	format	指定您的 CSV 输入文件是否符合 RFC-4180 要求。您可以输入以下任一值： • rfc4180 • simple 如果您输入 rfc4180，“csv-to-xml”步骤会假设您的 CSV 输入文件符合 RFC-4180 标准。 如果您输入 simple 或将该字段留空，则“csv-to-xml”步骤将执行以下操作： • 将撇号视为备用引号。 • 去除值开头的空格。 • 接受任何行结束符 (CR、LF 或 CRLF)。
使用第一行作为标题	useFirstLineAsHeader	指定“csv-to-xml”步骤是否使用 CSV 输入文件的第一行来命名 XML 输出文件中显示的列。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的“csv-to-xml”步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	该条件决定是否执行“csv 转 xml”步骤。

参考信息：Character-Conversion 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	职务申请中具有特征性转换的中介步骤的唯一编号。
输入	input	指定字符转换中介步骤从何处获取数据作为输入。共有 5 个可能的值：

属性A	XML 属性名称	描述
		• <i>message</i> : 整条消息, 包括 <i>rootpart</i> 和附件。 • <i>soapbody</i> : 中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i> : 消息中的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i> : MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件, 则系统会选中整封邮件。 • 变量 : 使用中 的已命名变量 <i>MediationContext</i>。
输出	output	指定 字符型转换 中介步骤将输出定向到何处。共有 5 个可能的值 : • <i>message</i> : 整条消息, 包括 <i>rootpart</i> 和附件。此值会创建一条新消息, 同时会移除所有附件。 • <i>soapbody</i> : 中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i> : 消息的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i> : MIME 消息根部分的内容, 如果消息不是 MIME, 则为消息本身。此值会保留原始消息中的所有附件。 • 变量 : 将输出另存为中的已命名变量 <i>MediationContext</i>。
移除重音符号	remove-accents	指定 字符转换 步骤在输出消息中是否将重音字符替换为非重音字符。 示例 : 如果设置为 <i>true</i>, 则 字符转换 步骤会将 <i>é</i> 替换为 <i>e</i>。
移除终止到期日期	remove-eols	指定 字符转换 步骤是否会移除输出消息中的行尾字符。
大写	uppercase	指定 字符转换 步骤是否将输出消息中的所有字符转换为大写。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 字符型转换 中介步骤进行描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	确定是否执行 字符转换 步骤的条件。
替换控件	replace-controls	指定 字符转换 步骤是否在输出消息中将 ASCII 控制字符替换为空格。

参考信息：Mtable-Builder 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集内的 mtable-builder 中介步骤的唯一编号。
输入	input	指定 mtable-builder 中介步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 rootpart 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。 • <i>变量</i>：使用中 的已命名变量 MediationContext。
表名称	mtable-name	的名称 MediationContext 属性，该存储将存储 mtable 对象。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 mtable-builder 中介步骤的描述。您可以

属性A	XML 属性名称	描述
		使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	该条件决定是否执行 mtable-builder 步骤。

参考信息：Mtable-Writer 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中的 mtable-编写器 中介步骤的唯一编号。
输出	output	指定 mtable-Writer 中介步骤将数据定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。 <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 <i>变量</i>：将输出另存为中的已命名变量 MediationContext。
表名称	mtable-name	的名称 MediationContext 属性，该存储将存储 mtable 对象。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 mtable-Writer 中介步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	该条件用于确定是否执行 mtable-编写器 步骤。

参考信息：ETV 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集内的 etv 中介步骤的唯一编号。
输入	input	指定 Etv 中介步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。 - 变量：使用中 的已命名变量 MediationContext。
输出	output	指定 Etv 中介步骤将输出定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 MediationContext。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对显示在合并日志中的 ETV 中介步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	此条件用于确定是否执行预估 更新 步骤。
附加消息	append-messages	指定新消息是替换现有消息，还是附加在现有消息之后。
消息属性	message-property	的名称 MediationContext 属性，用于存储 ETV 中介步骤创建的消息。

参考信息：XTT 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中的 xtt 中介步骤的唯一编号。
输入	input	指定 xtt 中介步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 rootpart 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。您必须指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容。如果邮件不是多用途互联网邮件扩展 (MIME) 类型的邮件，则系统会选中整封邮件。 • <i>变量</i>：使用中 的已命名变量 MediationContext。
输出	output	指定 xtt 中介步骤将输出定向到何处。共有 5 个可能的值： • <i>message</i>：整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。

属性A	XML 属性名称	描述
		<code>soapbody</code>：中 SOAP 正文的内容 <code>rootpart</code> 消息的“”。 <code>attachment</code>：消息的 MIME 附件。您必须指定附件索引。 <code>rootpart</code>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 <code>MediationContext</code>。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	<code>description</code>	对 xtt 中介步骤显示在合并日志中的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	<code>execute-when</code>	确定是否执行 xtt 步骤的条件。
附加消息	<code>append-messages</code>	指定新消息是替换现有消息，还是附加在现有消息之后。
消息属性	<code>message-property</code>	的名称 <code>MediationContext</code> 属性，用于存储 xtt 中介步骤创建的消息。

Mtable-Builder 子元素

概念：Mtable-Builder 子元素

“mtable-builder”步骤支持以下子元素：

子元素	描述
<code>csv-mtable-reader</code>	将 CSV 内容解析为 mtable 对象。
<code>json-mtable-reader</code>	将 JSON 内容解析为 mtable 对象。

参考信息：CSV-Mtable-Reader 属性

“Common”选项卡

属性A	XML 属性名称	描述
列名称	<code>column-names</code>	以逗号分隔的列名称列表，显示在您的 mtable 对象。

属性A	XML 属性名称	描述
包含标题行	contains-header-row	指明您的 mtable 对象包含表头行。
索引列	index-column	中显示的索引列的名称 mtable 对象。
使列名称标准化	normalise-column-names	指明 Workday 中的列名称 mtable 对象与空格一起显示。 如果设置为 true，上的列名称 mtable 对象与空格一起显示。 如果设置为 false，上的列名称 mtable 对象显示无空格。
分隔符	separator	用于分隔中的字段的分隔符 mtable 对象。
使用标题行	use-header-row	指明您的 mtable 对象使用标题行来显示列名称。

参考信息：JSON-Mtable-Reader 属性

“Common”选项卡

属性A	XML 属性名称	描述
解析目标	parse-target	要解析到数据中的根 JSON 对象或数组的名称 mtable 对象。
忽略编号列	ignore-id-column	指定 JSON 输入数据中的编号列是否显示 mtable 对象。 如果设置为 true，则 Workday 会忽略 Workday 中 JSON 输入数据中的编号列。mtable 对象。 如果设置为 false，Workday 会在您的 JSON 文件中显示“编号”列 mtable 对象。

Mtable-Writer 子元素

概念：Mtable-Writer 子元素

“mtable-writer”步骤支持以下子元素：

子元素	描述
csv-mtable-writer	显示 a 的内容 mtable 对象 (CSV 格式)。
json-mtable-writer	显示 a 的内容 mtable 对象 (CSV 格式)。

参考信息：CSV-Mtable-Writer 属性

“Common”选项卡

属性A	XML 属性名称	描述
列名称	column-names	以逗号分隔的列名称列表，显示在您的 mtable 对象。
行分隔符	line-separator	中的新行分隔符 mtable 对象。 支持以下值： • LF 适用于 Windows 和 Mac OS X。 • CR (Mac OS 版本 9 或更低版本) • CRLF 对于 Microsoft Windows。
分隔符	separator	用于分隔中的字段的分隔符 mtable 对象。
编写列名称	write-column-names	指明您的 mtable 对象使用标题行在中显示列名称 MediationContext。

参考信息：JSON-Mtable-Writer 属性

“Common”选项卡

属性A	XML 属性名称	描述
字段名称	field-name	显示搜索内容的 JSON 字段的名称 mtable 对象。
将数据表写入为对象	write-mtable-as-object	指示单行 Mtable 对象是否显示为： • 包含一个对象元素和多个字段的 JSON 数组 (True)。 • 具有多个字段的 JSON 对象 (False)。 默认设置为“false”。

日志记录和事件记录步骤

概念：日志记录和事件记录步骤

通过日志记录步骤，您可以创建自定义的日志消息，以便输出到不同的目标。Workday Studio 在“Palette”中提供以下日志记录和事件记录步骤：

名称	函数	注释
log	创建将写入 Workday 的自定义 Log4j 消息。	插入 MVEL 表达式时按“Ctrl + 空格键”可访问内容助手。
cloud-log	让您能够以 HTML 或 CSV 格式输出有关集成流程的自定义消息，然后将这些内容发送到您选择的端点。	“cloud-log”步骤会生成输出文件，但不会将其定向到任何位置。您需使用另一个步骤来处理文件的存储或传输。示例：使用“store”步骤将输出文件发送到您的租户。 “cloud-log”步骤生成的输出文件始终包含 5 列： - Timestamp - 级别 - 参考编号 - Message Summary - Message Details 使用“Extra Columns”选项卡可添加额外的信息。 输出文件的大小上限为 256 MB。如果您的输出文件超过 256 MB，系统通知您结果将被截断。

参考信息：Log 步骤属性

“Message Builder”选项卡

“Message Builder”简化了为某些程序集步骤定义消息内容的操作。

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中 日志 步骤的唯一编号。
输入	input	指定 记录 步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 <i>attachment</i>：消息中的 MIME 附件。指定附件索引。 <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件

属性A	XML 属性名称	描述
		不是 MIME，则 Workday 会选择整封邮件。 变量：使用中 的已命名变量 MediationContext。
级别	level	Log4J 日志记录级别。有以下 5 个选项： - debug - info - warn - error - fatal

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并的日志中显示的 日志 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	决定是否执行 日志 步骤的条件。
种类	category	记录器名称的一部分。如果您未指定值，Workday 会使用 DEFAULT。

您可以使用 `static-file` 子元素，用于为输入文件指定字符编码属性。示例：

```
<cc:log-message>
  <cc:static-file character-encoding="UTF-16" input-
file="logfile.html"/>
</cc:log-message>
```

如果您未指定 `character-encoding` 值，则 Workday 会使用指定的字符集。 `output-mimetype` 属性。如果此处也未指定字符集，Workday 将使用 UTF-8。

参考信息：Cloud-Log 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中的 云日志 步骤的唯一编号。
变量名称	variable-name	用于存储输出文件的变量的名称。您可以为多个“cloud-log”步骤指定相同的变量名称，以将它们的消息串联成一份报告。您可以使用不同的变量名称来生成单独的输出文件。

属性A	XML 属性名称	描述
级别	level	日志消息的严重程度有以下 6 个选项： • debug • info • warn • error • fatal • custom 如果您选择“custom”，则必须在相邻字段中使用 MVEL 表达式来指定自定义属性。
标头	header	云日志文件标头的文本。如果您将此字段留空，Workday Studio 会使用集成名称作为标头文本。 如果程序集有多个已定义标头的“cloud-log”步骤，Workday Studio 会使用流程中第一个步骤的标头。
消息	message	日志消息的文本。
消息详细信息	message-details	日志消息的描述。
参考编号	reference-id	Workday 对象的参考编号。用于问题排查。
条件	condition	一个 MVEL 布尔值表达式，指定 Workday 在什么情况下记录消息。 如果未指定条件，Workday 会始终记录消息。
输出文件类型	output-file-type	云日志 步骤生成的文件的类型。有以下 3 个选项： • HTML • CSV • XLSX 输出文件的大小上限为 256 MB。如果输出超过 256 MB，Workday Studio 会通知您并截断结果。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	日志消息的描述。
执行时间	execute-when	用于确定是否执行 云日志 步骤的条件。

“Extra Columns”选项卡

属性	XML 属性名称	描述
键	key	其他自定义消息的标识符。在 1 个日志变量中唯一。
标签	label	其他自定义消息中面向用户的标签。
值	value	其他自定义消息的文本。

relatedlinks

参考信息

[Workday 31 What's New Post: Workday Studio](#)

消息操作步骤

概念：消息操作步骤

通过消息操作步骤，您可以在消息通过您的集成时自定义消息的内容。Workday Studio 在“Palette”中提供以下消息操作步骤：

名称	函数	注释
copy	提取传入消息的各个部分，并将其复制到传出消息的不同部分。	
javascript	实现 JavaScript 脚本或函数，以对传入消息执行操作。	
set-headers	对中介消息的标头执行操作。	
write	使您能够构建自定义数据，并将其设置为中介消息的内容。	插入 MVEL 表达式时按“Ctrl + 空格键”可访问内容助手。

参考信息：Copy 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	复制 步骤在程序集中的唯一编号。
输入	input	指定 复制 步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 <i>attachment</i>：消息中的 MIME 附件。指定附件索引。

属性A	XML 属性名称	描述
		<i>rootpart</i> : 的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 - 变量：使用中 的已命名变量 <i>MediationContext</i>。
输入 XPath	<i>input-xpath</i>	一个 XPath 表达式，用于从消息部件中提取 XML 输入数据的匹配版块，或从路径中的变量中提取 XML 输入数据的匹配版块。 <i>MediationContext</i> 。
输出	<i>output</i>	指定 复制 中介步骤将输出定向到何处。共有 5 个可能的值： <i>message</i> : 整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i> : 中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 <i>attachment</i> : 消息的 MIME 附件。您必须指定附件索引。 <i>rootpart</i> : MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 <i>MediationContext</i>。
输出 XPath	<i>output-xpath</i>	一个 XPath 表达式，用于替换中消息部件或变量中 XML 输出数据的匹配版块。 <i>MediationContext</i> 。
附加到输出元素	<i>append-to-output-element</i>	指定 复制 步骤是否将输入 XML 元素附加到与 <i>output-xpath</i> 值
命名空间	<i>namespaces</i>	以下格式中 XPath 表达式使用的命名空间 <i>prefix namespace prefix2 namespace2</i> 。重设 <i>Assassing.xml</i> 文件中的命名空间映射。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	显示在合并日志中的 复制 步骤描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	确定是否执行 复制 步骤的条件。
流 XPath	stream-xpath	指定 复制 步骤是否为简单 XPath 表达式提供流式支持。
XPath 版本	xpath-version	指定使用 XPath 1.0 还是 2.0 版本。对于 Workday 11 及之前的程序集版本，请输入值 1；对于 Workday 12 及之后的程序集版本，请输入值 2。如果将“stream-xpath”值设置为“true”，则 Workday 会忽略此属性。 您可以允许 Workday 使用可插入系统属性值的 MVEL 模板动态设置版本。示例： @{props.get('the_version')}。

参考信息：JavaScript 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	JavaScript 步骤在程序集中的唯一编号。
输入	input	指定 JavaScript 步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 rootpart 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i>：的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。

属性A	XML 属性名称	描述
		变量：使用中 的已命名变量 MediationContext。
输出	output	指定 JavaScript 中介步骤将输出定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。 <i>attachment</i>：消息的 MIME 附件。您必须指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 MediationContext。
函数	function	包含 JavaScript 函数的文件名，该函数必须位于项目的 ws/WEAR-INF 目录中。 为“Function”或“Script”属性指定值，但不能同时为这两个属性指定值。您无法部署同时配置了这两个值的“javascript”步骤。
脚本	script	包含 JavaScript 脚本的文件名，该脚本必须位于项目的 ws/WEAR-INF 目录中。 为“Function”或“Script”属性指定值，但不能同时为这两个属性指定值。您无法部署同时配置了这两个值的“javascript”步骤。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 JavaScript 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	确定是否执行 JavaScript 步骤的条件。

参考信息：Set-Headers 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中设置了 标头的 步骤的唯一编号。
输出	output	指定 “set-headers” 中介步骤将输出定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 <i>attachment</i>：消息的 MIME 附件。指定附件索引。 <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 <i>MediationContext</i>。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	“设置标头”步骤的描述。显示在合并日志中。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	确定是否执行 已设置的表头 步骤的条件。
全部清除	clear-all	清除中介消息的所有标头。
清除所有排除项	clear-all-exclude	列表头名称，当将 “全部清除” 设置为“”时，不会删除这些名称。 true。

“标头”选项卡

Add Headers

属性	XML 属性名称	描述
名称	name	要添加到中介消息的标头的名称。

属性	XML 属性名称	描述
值	value	要添加到中介消息的标头的值。

Remove Headers

属性	XML 属性名称	描述
名称	name	要从中介消息中移除的标头的名称。

参考信息：Write 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中 写入 步骤的唯一编号。
输入	input	指定 写入 步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 • 变量：使用中 的已命名变量 <i>MediationContext</i>。
输出	output	指定 写入 中介步骤将输出定向到何处。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i>：消息的 MIME 附件。指定附件索引。 • <i>rootpart</i>：MIME 消息根部分的内容，如果消息不是 MIME，则为消息本身。此

属性A	XML 属性名称	描述
		值会保留原始消息中的所有附件。 变量：将输出另存为中的已命名变量 <code>MediationContext</code>。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	<code>description</code>	对合并日志中显示的 写入 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	<code>execute-when</code>	确定是否执行 写入 步骤的条件。

验证步骤

概念：验证步骤

通过验证中介步骤，您可以在消息通过您的集成时对其进行评估。Workday Studio 在“Palette”中提供以下验证步骤：

名称	函数	注释
<code>eval</code>	让您能够直接评估表达式。	
<code>validate</code>	验证中介消息，以确保它符合 XML 架构或文档类型定义 (DTD)，或者它包含格式正确的 XML。	- 验证失败会引发错误或放弃消息，具体取决于“Filter”属性的值。 - 您可以指定自定义错误消息。 - 您可以提供仅用于验证所选元素的架构或 DTD，并可以通过提供 XPath 表达式从中介消息中选择这些元素。
<code>validate-exp</code>	使用一个或多个 MVEL 表达式验证传入消息。	MVEL 表达式定义为子元素序列。
<code>validate-xpath</code>	使用 XPath 表达式验证消息，该表达式的求值结果为布尔值。	如果返回的 XPath 表达式不为空，则视为验证成功。

参考信息：Eval 步骤属性

“Expressions”选项卡

属性	XML 属性名称	描述
表达式	expression	让您能够直接输入 MVEL 条件表达式。 要在任何现有行下方添加新表达式，请单击“Add”，然后输入表达式。 要在现有行上方添加新表达式，请将光标放在该行中，然后单击“Add”下拉菜单中的“Insert”。 使用“Expressions”选项卡右侧的按钮可删除行或对行重新排序。

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	集合内 评估 步骤的唯一编号。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 评估 步骤进行描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	用于确定是否执行 评估 步骤的条件。
语言	language	指定要使用的表达式语言。MVEL 是当前唯一受支持的语言。

参考信息：Validate 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中 验证 步骤的唯一编号。
输入	input	指定验证 步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消

属性A	XML 属性名称	描述
		息的“”。仅当输入中包含 SOAP 消息时才适用。 • attachment：消息中的 MIME 附件。指定附件索引。 • rootpart：的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Studio 会选择整封邮件。 • 变量：使用中的已命名变量 MediationContext。
模式	schema	步骤用于验证消息的架构文件的位置。如果您指定位置而不是 MVEL 模板，Workday 将使用池化的验证程序。
命名空间	namespaces	以下格式中 XPath 表达式使用的命名空间 prefix namespace prefix2 namespace2。重设Assassing.xml 文件中的命名空间映射。
Xpath	xpath	一个 XPath 表达式，用于选择要验证的元素。
失败消息	failure-message	自定义错误消息，出现验证错误时，该消息将重设默认错误消息。仅当您将“Filter”属性设置为“false”后，错误消息才会显示。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 验证 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	确定是否执行 验证 步骤的条件。
模式	mode	执行的验证的类型。选项包括： • schema：根据 W3C XML 架构进行验证。 • dtd：根据文档类型定义进行验证。 • well-formed：验证 XML 格式是否正确。
DTD	dtd	该步骤用于验证消息的 DTD 文件的位置。

属性A	XML 属性名称	描述
替换消息	replace-message	指明当验证失败时，是否将设置的自定义错误消息用作“失败消息”属性。
XPath 版本	xpath-version	指定使用 XPath 1.0 还是 2.0 版本。对于 Workday 11 及之前的程序集版本，请输入值 1；对于 Workday 12 及之后的程序集版本，请输入值 2。如果将“stream-xpath”值设置为“true”，则 Workday 会忽略此属性。 您可以允许 Workday 使用可插入系统属性值的 MVEL 模板动态设置版本。示例： @{props.get('the_version')}}。
“筛选器”	filter	指定在验证失败时是否中断中介消息处理。默认为“false”，这意味着 Workday Studio 会将验证错误传递给构成中定义的错误处理程序。 如果设置为“true”，则 Workday Studio 会执行构成的响应步骤，但会放弃消息，并且不再调用任何步骤。这不会引发错误。

参考信息：Validate-Exp 步骤属性

“Expressions”选项卡

属性	XML 属性名称	描述
值	expression	让您能够直接输入 MVEL 条件表达式。 要在任何现有行下方添加新表达式，请单击“Add”，然后输入表达式。 要在现有行上方添加新表达式，请将光标放在该行中，然后单击“Add”下拉菜单中的“Insert”。 使用“Expressions”选项卡右侧的按钮可删除行或对行重新排序。
失败消息	failure-message	当验证表达式失败时，要添加到异常消息的文本。
错误编号	error-number	在验证表达式失败时返回的异常消息中添加一个错误编号。

属性	XML 属性名称	描述
替换	replace	表明是否使用“失败消息”属性中指定的文本替换 Workday 返回的失败消息。

公用选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中的“validate-exp”步骤的唯一编号。
“筛选器”	filter	指定在验证失败时是否中断中介消息处理。默认值为“false”，这意味着 Workday 会将验证错误传递给构成中定义的错误处理程序。 如果设置为“true”，则 Workday 会执行构成的响应步骤，但会放弃消息，并且不再调用任何步骤。这不会引发错误。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的“validate-exp”步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	一个条件，用于确定是否执行 validate-exp 步骤。

参考信息：Validate-Xpath 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中 validate-xpath 步骤的唯一编号。
输入	input	指定 validate-xpath 步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 <i>attachment</i>：消息中的 MIME 附件。指定附件索引。

属性A	XML 属性名称	描述
		<code>rootpart</code>：的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 - 变量：使用中 的已命名变量 MediationContext。
“筛选器”	<code>filter</code>	指定在验证失败时是否中断中介消息处理。默认值为“false”，这意味着 Workday 会将验证错误传递给构成中定义的错误处理程序。 如果设置为“true”，则 Workday 会执行构成的响应步骤，但会放弃消息，并且不再调用任何步骤。这不会引发错误。
命名空间	<code>namespaces</code>	以下格式中 XPath 表达式使用的命名空间 prefix namespace prefix2 namespace2。重设Assassing.xml 文件中的命名空间映射。
Xpath	<code>xpath</code>	一个 XPath 表达式，用于选择要验证的元素。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	<code>description</code>	在合并日志中显示的 validate-xpath 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	<code>execute-when</code>	此条件用于确定是否执行 validate-xpath 步骤。
XPath 版本	<code>xpath-version</code>	指定使用 XPath 1.0 还是 2.0 版本。对于 Workday 11 及之前的程序集版本，请输入值 1；对于 Workday 12 及之后的程序集版本，请输入值 2。如果将“stream-xpath”值设置为“true”，则 Workday 会忽略此属性。 您可以允许 Workday 使用可插入系统属性值的 MVEL 模板动态设置版本。示例： @{props.get('the_version')}。

编码步骤

概念：编码步骤

通过编码步骤，您可以对文件的内容进行编码、解码、压缩和解压缩。MediationContext。

Workday Studio 在“调色板”上提供了以下 编码 步骤：

名称	函数	注释
pgp-加密	使用 PGP 检索公钥，并加密出站数据。	- 要使用 Workday 中存储的密钥加密消息，请为您的集成配置传送服务。Workday Studio 集成无权访问 Workday 中存储的密钥。 - 您还可以使用 file:// 协议来引用本地 PGP 密钥，然后将其与集成一起部署。但是，对于 Workday Studio 集成而言，最好使用传送服务方法。
pgp-decrypt	检索私钥，并解密使用 PGP 加密的进站数据。	
base64-decode	将采用 Base64 编码的附件数据解码为二进制格式。	
base64-encode	使用 Base64 编码格式对二进制附件数据进行编码。	
zip	转换中的根部分和附件 MediationContext 压缩为一个 ZIP 文件。	
unzip	提取 ZIP 文件的内容。	如果 ZIP 中包含多个文件，则首选类似于中介消息的多部分结构。示例：如果您使用默认输出选项“message”，则 ZIP 中的第一个文件将成为根部分，而 ZIP 中的其他文件将成为附件。
压缩	转换中的单个文件 MediationContext 保存为 GZIP 或 BZIP2 文件。	与 ZIP 步骤不同，此步骤不支持文件存档。在压缩单个文件时，可使用此步骤。
decompress	提取单个 GZIP 或 BZIP2 文件的内容。	

概念：PGP 文件名

当使用 PGP 对纯文本文件进行加密时，可以为加密文件指定与原始文件不相关的文件名，去掉文件内容提示，从而确保安全性。但是，原始文件名仍会保留下来。它嵌入在加密文件中，以便稍后供收件人的解密系统检索。示例：您加密了一个名为 myPlaintext.txt 的文件。加密后的文件名为 myEncryption.pgp。

该文件的已加密内容中包含原始文件名。当您的收件人解密该文件时，他们的系统会将解密后的文件命名为“myPlaintext.txt”。

Workday 传送服务和检索服务

Workday 传送服务和检索服务会尽可能遵循常见的 PGP 文件命名惯例。

Workday 传送服务会在传出的 PGP 文件中嵌入纯文本文件名，以使相应文件在解密时可以恢复为该名称。

当 Workday 检索服务解密附加到集成事件的传入 PGP 文件时，它会使用嵌入的纯文本文件名来为解密后的文件命名。

但是，有时传入的 PGP 文件没有嵌入的纯文本文件名。在这种情况下，Workday 检索服务会根据 PGP 文件的名称来为解密后的文件命名：

- 如果 PGP 文件名以 .pgp 结尾，检索服务将从解密后的文件名中移除该后缀。示例：PGP 文件 myFile.csv.pgp 将解密为 myFile.csv，而 PGP 文件 myFile.pgp 将解密为 myFile。
- 如果 PGP 文件名以 .xlsx 结尾，检索服务将保留该文件名。示例：PGP 文件 myFile.xlsx 将解密为 myFile.xlsx。
- 在所有其他情况下，检索服务都会添加后缀“.plaintext”。示例：PGP 文件 myFile.doc 将解密为 myFile.plaintext。

Workday Studio

当您使用 Workday Studio 的“pgp-encrypt”步骤加密消息时，可以使用“Decrypted Filename”步骤属性指定纯文本文件名。然后，Workday Studio 会将该文件名嵌入到加密后的文件中，以便后续以常规方式检索。

同样，Workday Studio 的“pgp-decrypt”步骤会使用任意嵌入的纯文本文件名来为解密后的文件命名。若要在没有嵌入文件名的情况下提供文件名，您可以使用“Decrypted Filename”步骤属性。

问题排查

如果加密文件的远程方无意中将嵌入的纯文本文件名设置为非收件人系统要求的某个值，则可能会出现错误。由于 PGP 文件名与纯文本文件名完全无关，因此无法推导出后者。在这种情况下，收件人没有任何选项。

示例：某项检索服务被配置为检索和解密文件。运行时，它会收集名为“ExternalStudents.pgp”的 PGP 加密文件，但名为 _CONSOLE 的文件会附加到集成事件以供集成处理。在这种情况下，远程系统已将 _CONSOLE 设置为嵌入的纯文本文件名。因此 Workday 集成会将文件识别为 _CONSOLE。

为防止出现此类问题，唯一方法便是确保 PGP 文件的发件人了解如何正确嵌入所需的纯文本文件名。

参考信息：PGP-Encrypt 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中的 pgp-加密 步骤的唯一编号。
输入	input	指定 pgp-加密 步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 rootpart 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。

属性A	XML 属性名称	描述
		• <i>attachment</i> : 消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i> : 的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME, 则 Workday 会选择整封邮件。 • 变量 : 使用中 的已命名变量 MediationContext。
输出	output	指定 pgp-加密 步骤的定向输出。共有 5 个可能的值 : • <i>message</i> : 整条消息, 包括 rootpart 和附件。此值会创建一条新消息, 同时会移除所有附件。 • <i>soapbody</i> : 中 SOAP 正文的内容 rootpart 消息的“”。 • <i>attachment</i> : 邮件中的 MIME 附件。指定附件索引。 • <i>rootpart</i> : 的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME, 则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。 • 变量 : 将输出另存为中的已命名变量 MediationContext。
ASCII Armored 格式	ascii-armored	指示 pgp-加密 步骤生成的 PGP 数据采用的是 ASCII-连接格式 (Base64 的 PGP 版本) 还是纯二进制格式。ASCII Armored 格式会生成可方便用于电子邮件的文本 (包括页眉和页脚)。
证书	certificate	用于加密的证书, 由以下 URL 类型之一提供 : • 用于从租户检索公钥的附件 URL。必须遵循一定的模式 attachment:certificate/ WIDvalue, 其中 WIDvalue 是租户中存储的公钥的 Workday 编号。示例 : 您可以使用以下表单中的集成附件来配置参考信息 :

属性A	XML 属性名称	描述
		attachment:certificate/ @ {intsys.getAttributeReferenced Encryption Key', 'WID')}} 其中 PGP Encryption Key 是集 成属性的名称。 • 一个 mctx URL，用于 从集成存储了公钥的上 下文变量中检索公钥。 示例：mctx:vars/ MyPublicKey 其中 MyPublicKey 是上下文 变量的名称。 • 存储在程序集的 WEAR- INF 文件夹或其某个子文 件夹中的公钥的文件名。 示例：PublicKey.asc 或 /keys/ PublicKey.asc。
包含完整性检查	containing-integrity- check	指示 pgp-加密 步骤是否生成消 息签名以供接收方进行完整性检 查。
解密的文件名	decrypted-filename	解密的文件的名称。示例： yourname.plaintext.mail。Workday 会在哈希字符串中包含指定的 值。
兼容 Pgp26x	pgp26x-compatible	指示 pgp-加密 步骤生成的加 密数据是否向后兼容较早的 pgp26x 版本。
私钥密码	private-key-passphrase	指定私钥的密码。
签名私钥	signing-private-key	用于对消息进行签名的私钥（作 为文件名），使用 file:// 协 议。 注：Workday Studio 集成的最 佳做法是使用传送服务（而非 pgp-加密 步骤）来执行加密。
文本模式	textmode	指定 true，表明要加密的数据 为文本格式，并包含 <CR><LF> 行结尾。 注：如果您指定 true 对于不 包含 <CR><LF> 行结尾的数 据，Workday 会报告错误。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	显示在合并日志中的 pgp-加密 步骤的描述。您可以使用此功能

属性A	XML 属性名称	描述
		来描述该步骤的功能以及该步骤依赖的数据。
执行时间	execute-when	此条件决定是否执行 pgp-加密步骤。

relatedlinks

参考信息

[Workday 31 What's New Post: Workday Studio](#)

参考信息：PGP-Decrypt 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中的 pgp-decrypt 步骤的唯一编号。
输入	input	指定 pgp-decrypt 步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 <i>attachment</i>：消息中的 MIME 附件。指定附件索引。 <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 - 变量：使用中 的已命名变量 <i>MediationContext</i>。
输出	output	指定 pgp-decrypt 步骤将数据定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 <i>attachment</i>：邮件中的 MIME 附件。指定附件索引。

属性A	XML 属性名称	描述
		<code>rootpart</code> : 的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 MediationContext。
私钥	private-key	Workday 私钥的文件名或 URI。使用 <code>file://</code> 协议可将其指定为文件名。
密码	passphrase	指定私钥的密码。
解密的文件名	decrypted-filename	设置 content-disposition pgp-decrypt 步骤的输出中的属性，如果输出是变量，则为文件名。但是，如果 pgp-加密 步骤指定了解密的文件 名值，则该值优先。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	显示在合并日志中的 pgp-decrypt 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	确定是否执行 pgp-decrypt 步骤的条件。

参考信息：Base64-Decode 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中的 Base64 解码 步骤的唯一编号。
输入	input	指定 Base64-decode 步骤从何处获取数据作为输入。共有 5 个可能的值： <code>message</code> : 整条消息，包括 rootpart 和附件。 <code>soapbody</code> : 中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。

属性A	XML 属性名称	描述
		• <i>attachment</i> : 消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i> : 的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 • 变量 : 使用中 的已命名变量 MediationContext。
输出	output	指定 Base64 解码 步骤的定向输出。共有 5 个可能的值： • <i>message</i> : 整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 • <i>soapbody</i> : 中 SOAP 正文的内容 rootpart 消息的“”。 • <i>attachment</i> : 邮件中的 MIME 附件。指定附件索引。 • <i>rootpart</i> : 的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。 • 变量 : 将输出另存为中的已命名变量 MediationContext。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 Base64 解码 步骤进行了描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	此条件决定是否执行 Base64 解码 步骤。

参考信息：Base64-Encode 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中的 Base64 编码 步骤的唯一编号。
输入	input	指定 Base64 编码 步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 <i>attachment</i>：消息中的 MIME 附件。指定附件索引。 <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 - 变量：使用中 的已命名变量 MediationContext。
输出	output	指定 Base64 编码 步骤的定向输出。共有 5 个可能的值： <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 <i>attachment</i>：邮件中的 MIME 附件。指定附件索引。 <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 MediationContext。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的步骤 进行 Base64 编码 的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	用于确定是否执行 Base64 编码 步骤的条件。

参考信息：Zip 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中 Zip 步骤的唯一编号。
输入	input	指定 压缩 步骤从何处获取数据 作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 • 变量：使用中 的已命名变量 <i>MediationContext</i>。
输出	output	指定 zip 步骤将输出定向到的位置。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i>：邮件中的 MIME 附件。指定附件索引。 • <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday

属性A	XML 属性名称	描述
		会选择整封邮件。此值会保留原始消息中的所有附件。 变量：将输出另存为中的已命名变量 MediationContext。
格式	format	压缩文件的格式。选项包括： - zip - tar 注: TAR 格式支持存档，但不支持压缩。要压缩 TAR 文件，请使用以下任一命令： - gzip，此命令会将 .tar 文件扩展名替换为 .tar.gz 或 .tgz - bzip2，此命令会将 .tar 文件扩展名替换为 .tar.bz2 您可以分别使用“compress”步骤和“decompress”步骤来压缩和解压缩在“zip”步骤中生成的 TAR 存档。
级别	level	0 到 9 范围内的压缩级别，其中 9 表示最大压缩。默认值为 -1。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	该 Zip 步骤的描述。显示在合并日志中。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	一个用于确定 Zip 步骤是否执行的条件。
备注	comment	嵌入到 ZIP 文件中的备注。备注是任意 ASCII 字符串，最多可包含 65,535 个字节。

参考信息：Unzip 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中解压缩步骤的唯一编号。

属性A	XML 属性名称	描述
输入	input	指定 解压缩 步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 • 变量：使用中 的已命名变量 MediationContext。
输出	output	指定 解压缩 步骤将输出定向到的位置。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i>：邮件中的 MIME 附件。指定附件索引。 • <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。 • 变量：将输出另存为中的已命名变量 MediationContext。
格式	format	压缩文件的格式。选项包括： • zip • tar 注： TAR 格式支持存档，但不支持压缩。要压缩 TAR 文件，请使用以下任一命令：

属性A	XML 属性名称	描述
		- gzip，此命令会将 .tar 文件扩展名替换为 .tar.gz 或 .tgz - bzip2，此命令会将 .tar 文件扩展名替换为 .tar.bz2 您可以分别使用“compress”步骤和“decompress”步骤来压缩和解压缩在“zip”步骤中生成的 TAR 存档。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	解压缩操作的描述MediationContext。
执行时间	execute-when	确定是否执行 解压缩 步骤的条件。
文件名	filename	解压缩内容时创建的文件的名称。

参考信息：Compress 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中 压缩 步骤的唯一编号。
输入	input	指定 压缩 步骤从何处获取数据作为输入。共有 5 个可能的值： - message：整条消息，包括 rootpart 和附件。 - soapbody：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 - attachment：消息中的 MIME 附件。指定附件索引。 - rootpart：的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 - 变量：使用中的已命名变量 MediationContext。
输出	output	指定 压缩 步骤将输出定向到何处。共有 5 个可能的值：

属性A	XML 属性名称	描述
		<i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 <i>attachment</i>：邮件中的 MIME 附件。指定附件索引。 <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 <i>MediationContext</i>。
格式	<i>format</i>	压缩文件的格式。选项为“gzip”和“bzip2”。默认格式为“gzip”。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	<i>description</i>	对合并日志中显示的 压缩 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	<i>execute-when</i>	确定是否执行 压缩 步骤的条件。
文件名	<i>filename</i>	解压缩内容时创建的文件的名称。

参考信息：Decompress 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	<i>id</i>	程序集内 解压缩 步骤的唯一编号。
输入	<i>input</i>	指定 解压缩 步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。

属性A	XML 属性名称	描述
		• <i>soapbody</i> : 中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i> : 消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i> : 的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME, 则 Workday 会选择整封邮件。 • 变量 : 使用中的已命名变量 <i>MediationContext</i>。
输出	output	指定 解压缩 步骤将输出定向到的位置。共有 5 个可能的值 : • <i>message</i> : 整条消息, 包括 <i>rootpart</i> 和附件。此值会创建一条新消息, 同时会移除所有附件。 • <i>soapbody</i> : 中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i> : 邮件中的 MIME 附件。指定附件索引。 • <i>rootpart</i> : 的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME, 则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。 • 变量 : 将输出另存为中的已命名变量 <i>MediationContext</i>。
格式	format	压缩文件的格式。选项为“gzip”和“bzip2”。默认格式为“gzip”。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 解压缩 步骤进行的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	此条件决定是否执行 解压缩 步骤。

属性A	XML 属性名称	描述
文件名	filename	解压缩内容时创建的文件的名称。

其他步骤

概念：其他步骤

通过步骤，您可以在消息通过您的集成时对其执行各项操作。Workday Studio 在“Palette”中提供以下其他步骤：

名称	函数
custom	使您能够将使用 Spring Bean 编写的自定义代码应用于传入消息。
set-dynamic-endpoint	为 out-transport 配置 WS-Addressing (WSA) 端点信息。
store	创建文档的永久副本，该副本可供其他用户使用，以及/或者可作为集成事件的附件进行查看。 “store”步骤单个消息的大小不得超过 1 GB（压缩后）。累计不得超过 3 GB（压缩后）。Workday 会终止超出上述限制的集成。请尽可能使用拆分器或流式传输来拆分集成中的大型文件。
retrieve	检索先前由“store”步骤存储的文档的永久副本。
xmldiff	比较 2 个 XML 输入文档。
enqueue-message	将消息整理到队列中。

参考信息：Custom 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	指定程序集中自定义步骤的唯一编号。
输入	input	指定自定义步骤从何处获取数据作为输入。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 <i>attachment</i>：消息中的 MIME 附件。指定附件索引。 <i>rootpart</i>：的内容 rootpart MIME 消息的

属性A	XML 属性名称	描述
		电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 变量：使用中的已命名变量 <code>MediationContext</code>。
输出	output	指定 自定义 步骤将输出定向到的位置。共有 5 个可能的值： <code>message</code>：整条消息，包括 <code>rootpart</code> 和附件。此值会创建一条新消息，同时会移除所有附件。 <code>soapbody</code>：中 SOAP 正文的内容 <code>rootpart</code> 消息的“”。 <code>attachment</code>：邮件中的 MIME 附件。指定附件索引。 <code>rootpart</code>：的内容 <code>rootpart</code> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 <code>MediationContext</code>。
方法名称	method-name	您的自定义 Java 类中的 Java 方法的名称。
java - java - java	ref	由 自定义 步骤实施的自定义 Spingle Java 类。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 自定义 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	确定是否执行 自定义 步骤的条件。

参考信息：Set-Dynamic-Endpoint 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集内的 set-Dynamic-Endpoint 步骤的唯一编号。
端点	endpoint	动态端点的地址。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 “Set-Dynamic-Endpoint” 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	该条件决定是否执行 “Set-Dynamic-Endpoint” 步骤。

参考信息：Store 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	程序集中 存储 步骤的唯一编号。
输入	input	指定 存储 步骤从何处获取数据作为输入。共有 5 个可能的值： • <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。 • <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i>：消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 • 变量：使用中 的已命名变量 MediationContext。
输出	output	指定 存储 步骤将输出定向到的位置。共有 5 个可能的值：

属性A	XML 属性名称	描述
		• <i>message</i> : 整条消息, 包括 <i>rootpart</i> 和附件。此值会创建一条新消息, 同时会移除所有附件。 • <i>soapbody</i> : 中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 • <i>attachment</i> : 邮件中的 MIME 附件。指定附件索引。 • <i>rootpart</i> : 的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME, 则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。 • 变量 : 将输出另存为中的已命名变量 <i>MediationContext</i>。 存储 步骤会将消息的 MIME 类型与消息本身一起存储。示例 : <i>text/plain</i>。使用“retrieve”步骤检索消息时, Workday 会自动使用已存储的 MIME 类型对其进行标记, 以便后续步骤可以正确处理该消息。 “store”步骤单个消息的大小不得超过 1 GB (压缩后)。累计不得超过 3 GB (压缩后)。Workday 会终止超出上述限制的集成。请尽可能使用拆分离器或流式传输来拆分集成中的大型文件。
收款	<i>collection</i>	存储您的文档的集合的名称。
过期时间	<i>expiresIn</i>	距离文档过期的时间段。 请以 XSD 持续时间格式指定值。 示例 : 如要指明 1 年 3 个月 5 天 8 小时 45 分 10 秒, 请输入 <i>P1Y3M5DT8H45M10S</i>。
汇总	<i>summary</i>	文档的摘要。在您的 Workday 租户中显示。
标题	<i>title</i>	文档的标题
创建文档参考信息	<i>createDocumentReference</i>	指定 存储 步骤是否自动调用 Workday 来创建文档资历证明。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 存储 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	该条件用于确定是否执行 存储 步骤。
内容处置	contentDisposition	文档的 Content-Disposition 标头。 重设“title”属性值（如已指定）。
二进制大型对象库 URI	blobitoryURI	的 URI cc-blobitory 的 Workday 租户中实施的服务。 示例：https://machine-name.workday.com:8080/ccx/cc-blobitory。
过期日期	expiresOn	文档到期的日期和时间。 重设 “Expires In”属性值（如果已指定）。 在中输入此属性的值 CCYY-MM-DDThh:mm:ssZ 格式。 示例：要指明 2009 年 12 月 31 日晚上 11 点 59 分，请输入 2009-12-31T23:59:00Z。
模式	schema	引用您的文档的架构的 URI。以下是 2 个可能的值： - http://www.w3.org/2005/Atom - urn:com.workday/bsvc/blob
条目编号	entryID	为每个文档指定“条目编号”。这在您的租户中必须唯一。如果您未指定条目编号值，Workday 会生成一个。如果您指定的编号值已存在，则 Workday 会为文档重设此值（前提是该文档可变更）。如果现有文档不可变更，则您无法重复使用条目编号。 在使用 REST API 更新文档时： - 对具有预定义条目编号的文档使用 PUT HTTP 方法。 - 如果未指定条目编号，或者条目编号的计算结果

属性A	XML 属性名称	描述
		为“空”或者空字符串，请使用 HTTP POST 方法。
是附件	isAttachment	指定您的文档是否为 OMS 附件。
不可更改	immutable	指定您是否可以在创建文档后对其进行修改。
指定所有者	assignOwner	指定 Workday 是否将您指定为文档的所有者。
文档参考描述	documentReferenceDescription	文档描述。在您的 Workday 租户中显示。

参考信息：Retrieve 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	指定程序集中 检索 步骤的唯一编号。
输出	output	指定 检索 步骤将输出定向到的位置。共有 5 个可能的值： <i>message</i>：整条消息，包括 <i>rootpart</i> 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 <i>rootpart</i> 消息的“”。 <i>attachment</i>：邮件中的 MIME 附件。指定附件索引。 <i>rootpart</i>：的内容 <i>rootpart</i> MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 <i>MediationContext</i>。 当您在业务流程中配置文档检索服务时，Workday 会使用您在“自定义内容类型”字段中输入的字符串作为该服务存储的文档的 MIME 类型。 当您使用“retrieve”步骤（直接或通过文档迭代器、文档访问器帮助程序对象）检索文档

属性A	XML 属性名称	描述
		时，Workday 会自动使用已存储的 MIME 类型来标记文档，以便后续步骤可以正确处理文档。如果您未输入自定义的内容类型，Workday 将使用 UTF-8 编码。为避免错误，请始终为非 UTF-8 编码的数据输入相应的自定义内容类型。
收款	collection	存储您的文档的集合的名称。
输入	entry	集合中文档的标识符。可从检索 MediationContext 使用 MVEL 表达式。 示例： @{parts[0].xpath('//blob:entry', 'bloburn:com.workday/bsvc/blob')}
密码	password	输入您的文档访问凭据的密码。
用户名	username	用于访问文档的用户凭据。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 检索 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	用于确定是否执行 检索 步骤的条件。

参考信息：Xmldiff 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	指定 xmldiff 步骤在程序集中的唯一编号。
第一个输入	first-input	指定 xmldiff 步骤的第一个数据输入源。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。

属性A	XML 属性名称	描述
		• <i>attachment</i> : 消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i> : 的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 • 变量 : 使用中 的已命名变量 MediationContext。
第二个输入	second-input	为 xmldiff 步骤指定第二个数据输入源。共有 5 个可能的值： • <i>message</i> : 整条消息，包括 rootpart 和附件。 • <i>soapbody</i> : 中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 • <i>attachment</i> : 消息中的 MIME 附件。指定附件索引。 • <i>rootpart</i> : 的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。 • 变量 : 使用中 的已命名变量 MediationContext。
输出	output	指定 xmldiff 步骤将输出定向到何处。共有 5 个可能的值： • <i>message</i> : 整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 • <i>soapbody</i> : 中 SOAP 正文的内容 rootpart 消息的“”。 • <i>attachment</i> : 邮件中的 MIME 附件。指定附件索引。 • <i>rootpart</i> : 的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。

属性A	XML 属性名称	描述
		变量：将输出另存为中的已命名变量 <code>MediationContext</code>。
复制到输出	<code>copy-to-output</code>	指定 <code>xmldiff</code> 步骤是将输出复制到中介消息还是变量。 如果设置为“true”，则“<code>xmldiff</code>”步骤会将输出复制到中介消息。 如果设置为“false”，则“<code>xmldiff</code>”步骤会将输出复制到变量。
忽略属性	<code>ignore-attributes</code>	指定 <code>xmldiff</code> 步骤在比较 2 个 XML 文档时是否忽略属性。
命名空间	<code>namespaces</code>	命名空间重设由 <code>xmldiff</code> 步骤实施的 XPath 表达式的定义。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	<code>description</code>	对合并日志中显示的 <code>xmldiff</code> 步骤的描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	<code>execute-when</code>	确定是否执行 <code>xmldiff</code> 步骤的条件。
XPath 版本	<code>xpath-version</code>	指定使用 XPath 1.0 还是 2.0 版本。对于 Workday 11 及之前的程序集版本，请输入值 1；对于 Workday 12 及之后的程序集版本，请输入值 2。如果将“<code>stream-xpath</code>”值设置为“true”，则 Workday 会忽略此属性。 您可以允许 Workday 使用可插入系统属性值的 MVEL 模板动态设置版本。示例： <code>@{props.get('the_version')}</code>

“定义”选项卡

属性	XML 属性名称	描述
名称	<code>name</code>	指定的名称 <code>sub-diff</code> 在查找 <code>xmldiff</code> 步骤的结果时使用。
忽略删除操作	<code>ignore-deletions</code>	指定 <code>xmldiff</code> 步骤是否忽略输出中的删除项。
仅限不同节点	<code>result-document-different-nodes-only</code>	指定在 <code>xmldiff</code> 步骤的输出文档中显示哪些节点。

属性	XML 属性名称	描述
		如果设置为 true，则在输出文档中仅显示顶层节点。 如果设置为 false，则所有子节点都显示在输出文档中。
包含结果文档	result-document-inclusion	指定 xmldiff 步骤是否记录输出文档中节点之间的差异。以下是 3 个可能的值： - Always：所选节点及其子节点始终显示在结果文档中。 - Different：仅在发现差异时，所选节点及其子节点才会显示在结果文档中。 - Never：所选节点及其子节点从不显示在结果文档中。
XPath	xpath	由 xmldiff 步骤实施的 XPath 表达式，用于在输入文档中查找节点。

“Matcher XPath”选项卡

在“Matcher XPath”选项卡上，您可以添加 XPath 表达式，用于查找输入文档中不连续的匹配节点。

参考信息：Enqueue-Message 步骤属性

“Common”选项卡

属性A	XML 属性名称	描述
步骤编号	id	集合中的 enqueue-message 步骤的唯一编号。
输入	input	指定 enqueue-message 步骤从何处获取数据作为输入。共有 5 个可能的值： - message：整条消息，包括 rootpart 和附件。 - soapbody：中 SOAP 正文的内容 rootpart 消息的“”。仅当输入中包含 SOAP 消息时才适用。 - attachment：消息中的 MIME 附件。指定附件索引。 - rootpart：的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。

属性A	XML 属性名称	描述
		变量：使用中 的已命名变量 MediationContext。
输出	output	指定 enqueue-message 步骤将输出定向到何处。共有 5 个可能的值： <i>message</i>：整条消息，包括 rootpart 和附件。此值会创建一条新消息，同时会移除所有附件。 <i>soapbody</i>：中 SOAP 正文的内容 rootpart 消息的“”。 <i>attachment</i>：邮件中的 MIME 附件。指定附件索引。 <i>rootpart</i>：的内容 rootpart MIME 消息的电子邮件地址。如果邮件不是 MIME，则 Workday 会选择整封邮件。此值会保留原始消息中的所有附件。 - 变量：将输出另存为中的已命名变量 MediationContext。
内容类型	contentType	重设输出消息的内容类型（默认为 text/xml）。
队列名称	queueName	enqueue-message 步骤将输出定向到的队列的名称。
文件名	fileName	重设中介消息标头中的文件名。

“Advanced”选项卡

属性A	XML 属性名称	描述
描述	description	对合并日志中显示的 排队消息步骤进行描述。您可以使用此功能来描述该步骤的功能以及该步骤所处理的数据。
执行时间	execute-when	此条件决定是否执行 消息 排队步骤。

程序集模块

创建程序集模块

开始之前
创建一个程序集项目。

关于此任务
Workday Studio 程序集模块提供用于构建程序集的参数化模板。这些模块让您能够创建和重复使用多组预先配置的构成和步骤，以执行特定任务。

- 过程
- 1. 添加要包括在程序集模块中的元素，并按需配置其属性。您可以基于功能完整的程序集创建模块，也可以基于较少的构件和步骤集来创建模块。用户可在实现模块时调整参数。
 - 2. 保存程序集。
 - 3. 在“Project Explorer”中，右键单击该程序集项目，然后选择“Create Assembly Module...”。
 - 4. 提供模块的名称和描述。描述将显示为模块的工具提示。
 - 5. 为模块选择一个图标。您可以使用默认图标，也可以上传您自己的图标。
 - 6. Workday Studio 会将模块参数整理到称为页面的集合中。一个参数必须属于某个页面。每个页面都有名称和描述。每个参数都有名称、标签（在“Configure Module”向导中显示），以及描述、类型和默认值。有 6 种参数类型：

类型	描述	示例
字符串	文本值。	"HelloWorld"
XMLRequest	请求消息的 XML 文本。模块用户使用“创建文字 XML”或“创建 Workday Web 服务请求”向导为 Web 服务请求指定值。	<?xml version="1.0" encoding="UTF-8"?> <env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <env:Body> <wd:Get_Workers_Request xmlns:wd="urn:com.workday/bsvc" wd:version="v23.2"> <wd:Request_References wd:Skip_Non_Existing_Instances="t <wd:Worker_Reference> <wd:ID wd:type="Employee_ID">abcdef</wd:ID> </wd:Worker_Reference> </wd:Request_References> </wd:Get_Workers_Request>

类型	描述	示例
		</env:Body> </env:Envelope>
提示	您可以使用类报告字段 (CRF)，为特定类型的 Workday 对象的启动参数提供提示。要指定类报告字段引用，请单击“Select reference id”。 注：要指定类报告字段参考，您必须连接到 Workday 租户。	fbbe44a681251000039acc2ff657000a
文件	文件名您可以将文件扩展名指定为 data 值	*.xslt
枚举	一个字面值或多个以逗号或空格分隔的值。	code1 code2 code3
布尔值	布尔值。	true

7. 在您添加完参数并保存模块后，Workday 会在 XML 编辑器中打开其 XMI 文件。该文件包含 params 元素来表示您指定的每个参数。该 assemblyText 与 diagramText 包含程序集 XML 和图表详细信息的元素，封装在 <![CDATA[]]> 标记。您可以直接编辑 XMI 文件。
8. 关闭该 XMI 文件。

将程序集模块添加到项目

开始之前

创建一个程序集项目或打开一个现有的程序集项目。

关于此任务

Workday Studio 程序集模块提供用于构建程序集的参数化模板。这些模块让您能够创建和重复使用多组预先配置的构成和步骤，以执行特定任务。

- 过程
1. 将模块从“Palette”的“Modules”版块拖动到 Assembly Editor 中。Workday Studio 将打开“Configure Module”向导。
- 注：“Module Library”仅显示您在以下首选项页面上选择的程序集模块：“Window”>“Preferences”>“Workday”>“Assembly Editor”>“Modules”。
2. 配置模块的参数。
3. 如果程序集模块包含“”类型的参数 XMLRequest，单击“创建文本 XML”或“创建 Workday Web 服务请求”，指定 Web 服务请求的值。完成此任务时，请参考以下信息：

向导	运营	注释
创建文字 XML	Get, Put	• 可用于任何 Web 服务。 • 从文本窗格下方的列表中，为请求选择 Web 服务名称和 Web 服务版本值。

向导	运营	注释
创建 Workday Web 服务请求	Get	- 仅用于 Workday Web 服务。 - 当您选择运算时，Workday Studio 会自动填充 Web 服务名称和版本值。

4. Workday Studio 会显示模块包含的程序集元素的预览。在您完成向导后，Workday Studio 会将这些元素添加到程序集中。

概念：程序集模块

Workday Studio 程序集模块提供用于构建程序集的参数化模板。这些模块让您能够创建和重复使用多组预先配置的构成和步骤，以执行特定任务。您可以根据整个程序集或其中的一部分来创建模块。

您使用模块构建的程序集与您从头开始构建的程序集没有区别。它们具有相同的外观和行为。

Workday Studio 会将模块保存为“XML 元数据交换”(XMI) 格式的模板，文件扩展名为 xmi。在 Workday Studio 的“Palette”中，通过“Module Library”即可访问您已保存的模块。要允许其他开发者访问您的模块，请将模块保存在共享位置。

通过共享经过反复验证的程序集模块，可帮助加快集成开发的速度并降低其成本。

“Module Library”仅显示您在以下首选项页面上选择的程序集模块：“Window” > “Preferences” > “Workday” > “Assembly Editor” > “Modules”。

子程序集

创建子程序集

开始之前

创建一个程序集。

关于此任务

子程序集是一种处理链，其第一个构成是“local-in”传输。您可以从同一程序集或另一个程序集中，通过“local-out”传输来调用子程序集。

过程

1. 添加并配置“local-in”传输，以定义子程序集的开头。
 - a) 在您希望转换为子程序集的处理链的开头，添加一个“local-in”传输。
 - b) 在该传输的“Properties”视图的“Common”和“Advanced”选项卡上，配置其属性。您可以为子程序集提供图标，以及作为工具提示显示的文本。
 - c) 在“Parameters”选项卡上，定义子程序集需要用作输入的参数。您可以在调用子部件号的本地输出传输上配置这些参数。Studio 将它们存储为 MediationContext 属性。
 - d) 在“OutParameters”选项卡上，定义 MediationContext 子部件设置的属性。
 - e) 保存该部件。

2. 添加并配置对子程序集进行调用的“local-out”传输。

- 向调用子程序集的程序集添加一个“local-out”传输。
- 在该传输的“Properties”视图的“Common”和“Advanced”选项卡上，配置其属性。指定子程序集本地输入 作为 端点 属性。使用以下格式：vm://assembly-name/local-out-ID，或单击“在子部件号中选择本地文字”。
- 在“Parameters”选项卡上，设置 MediationContext 属性。您可以通过单击“创建‘设置’元素”创建新参数。您还可以选择您在“local-in”传输中定义的任何参数，并通过单击“Select Parameter”为这些参数赋值。

注：在“Select Parameters”向导中，星号表示属性为必需。问号表示用途变化不定。

- 保存该程序集。

概念：子程序集

子程序集是一种处理链，其第一个构成是“local-in”传输。“local-in”定义了处理链的入口点，使其能够由同一程序集或其他程序集中的“local-out”传输进行调用。您可以在同一程序集中包含多个子程序集和“local-out”传输。

注：您不能单独将子程序集部署到 Workday 中，因为子程序集并非以“workday-in”传输开头。

您可以随意调用子程序集，因此您可以封装处理逻辑，以供重复使用。您可以将它们视为子例程。local-in 传输的默认编号为 SubRoutine。本地输出 传输类型的默认编号为 CallSubRoutine。

子装配件具有参数。您在 本地输入 传输上指定它们，并在 本地输出 上为它们指定值。以下参数定义中的属性 MediationContext。

当您保存子装配件时，Workday Studio 会自动将调用该子装配件的 本地转出 传输器添加到 选用板中的“工作区组件”种类。如果您在子程序集的“local-in”传输中提供了自定义图标，则此“local-out”传输将使用该自定义图标。如果您未提供，则该传输将使用问号图标。

“Palette”会始终显示公开子程序集，但对于私有子程序集，仅当其出现在程序集中时才会显示。Workday Studio 在“Palette”的“Common Components”种类中提供了多个预封装的子程序集。

要找到“local-out”引用的“local-in”传输，请右键单击“local-out”，然后选择“Declaration in Workspace”。

要找到由“local-in”引用的“local-out”传输，请右键单击“local-in”，然后选择“References in Workspace”。

relatedlinks

概念

[概念：通用构成](#) on page 81

自定义程序集构成

创建自定义 Java 类和 Spring Bean

开始之前

创建一个程序集项目。

关于此任务

在 Workday Studio 中，您可以通过“Palette”创建 4 个自定义程序集元素：

- “custom-mediation”构成
- “custom”步骤
- “custom-out”传输
- custom-error-handler

第五个自定义元素“custom routing strategy”可从“Route”构成的上下文工具栏中获取。

过程

1. 在 Assembly Editor 中打开一个程序集。
2. 将以下自定义元素之一添加到该程序集中：
 - “custom-mediation”构成
 - “custom”步骤
 - “custom-out”传输
 - custom-error-handler
3. 在自定义元素的“Properties”视图中，配置其属性。
4. 单击“Spring Bean”字段右侧的“Create Spring Bean Properties and Java Class”。Workday Studio 显示的属性会因自定义元素类型而略有不同。完成此任务时，请参考以下信息：

选项	描述
Id	提供唯一的 Spring Bean 识别号。
范围	• 选择“singleton”可随每项引用返回相同的 Bean 实例。 • 选择“prototype”可随每项引用返回不同的 Bean 实例。
Package	指定 Java 封装。
名称	指定类名称。
Input Type	选择要传递给 Java 方法的输入类型。您无需指定实现详细信息，即可访问消息部分并将其转换为相应的类型。Workday 会检查该方法的第一个参数，并将消息部分转换为相同的类型。
Output Type	选择方法的输出（返回）类型。
Method Name	输入要在 Java 类中定义的 Java 方法的名称。默认值为 process。
包括动态端点的参数	指定是否包括动态端点的参数。
Annotate Java class to appear in palette	选中此复选框可在“Palette”中显示自定义元素。此选项会向生成的 Java 类添加注释，该类位于程序集项目文件夹的“src”文件夹中。
Tooltip	输入显示为自定义元素描述的文本。

5. 保存该程序集。

结果

Workday Studio 会生成表示指定封装内自定义元素的 Java 类。它还会在 assembly.xml 文件底部添加对新的自定义 Spring Bean 的引用。

下一步做什么

将所需代码添加到您的自定义 Java 类。

概念：自定义程序集元素

在 Workday Studio 中，您可以创建 5 种类型的自定义程序集元素：

自定义元素	程序
中介构成	通过“Palette”的“Components”种类添加“custom-mediation”构件。
中介步骤	通过“Palette”的“Other”种类添加“custom”步骤。
出站传输	通过“Palette”的“Out Transports”种类添加“custom-out”传输。
错误处理程序	通过“Palette”的“Error Handlers”种类添加“custom-error-handler”构成。
路由策略	通过“Route”构成的上下文工具栏，选择“Create custom-strategy”。

在每种情况下，自定义元素均包含 3 个部分：

- 基础 Java 实现，定义了元素所用的类和方法。
- Spring Bean 配置，用于向 Workday 说明如何创建 Java 类实例。
- 程序集构成配置，用于将您的自定义元素置于图形化程序集视图中。

在处理传入数据时，自定义程序集元素可按常规方式访问消息部分。您还可以指定 Workday 应将自定义元素的输出定向到消息的哪个部分。

要共享自定义元素，请将 Java 包导出为二进制 JAR 文件。然后，其他开发者可将该 JAR 添加到其项目的类路径中。

自定义步骤

在为 自定义 步骤编写 Java 实施时，请参考以下几点：

- 必须声明该方法 `public` 这样就可以从类的外部调用该函数。
- 方法必须是非静态方法。
- 每个方法的名称都必须唯一。
- 参数列表必须与所支持的组合匹配。
- 方法可引发任何类型的异常。
- 如果您未提供方法名称，Workday 会选择其参数列表与受支持的方法签名匹配的任何公共方法（忽略 getter 和 setter）。如果有任何不清晰之处，Workday 会在部署期间报告错误，您可以通过指定方法名称来解决该错误。

“custom”步骤必须提供符合以下某个签名的方法：

签字	用途	示例
<code>public return type method(input parameter type input) [throws any Exception]</code>	定义方法的返回类型和输入参数。	<code>public String addTimestamp(byte[] messageContent) throws IOException {</code>
<code>public void method(input parameter type input, output parameter type output) [throws any Exception]</code>	定义受支持的输入和输出参数。	<code>public void addTimestamp(byte[] messageContent, OutputStream out) throws IOException {</code>

自定义 步骤支持以下输入、返回和输出参数类型：

输入类型	返回类型	输出参数类型
• <code>java.lang.String</code>	• <code>void</code>	• <code>java.io.OutputStream</code>

输入类型	返回类型	输出参数类型
• byte[] • java.io.InputStream • javax.activation.DataHandler • javax.activation.DataSource • org.w3c.dom.Element • org.w3c.dom.Document • javax.xml.transform.Source • javax.xml.transform.stream.StreamSource • javax.xml.transform.dom.DOMSource	• boolean (仅限自定义步骤) • java.lang.String • byte[] • java.io.InputStream • javax.activation.DataHandler • javax.activation.DataSource • org.w3c.dom.Element • org.w3c.dom.Document • javax.xml.transform.Source	• javax.xml.transform.Result

如果 自定义 步骤返回 void 类型或 null 值，则此设置不会更改中介消息。

概念：自定义元素封装

Java 类注释

如果您想在不同的项目中重复使用某个自定义程序集元素，请使用注释使该元素显示在“Palette”中。

要执行此操作，请在创建 Java 类和 Spingle bean 时选中“注释 Java 类以出现在面板中”复选框。Studio 添加了 @Component 在生成的 Java 类中添加“”注释，该类位于程序集项目的 src 文件夹中。示例：

```
import com.capeclear.assembly.annotation.Component;
import com.capeclear.assembly.annotation.ComponentMethod;
import com.capeclear.assembly.annotation.Property;
import static com.capeclear.assembly.annotation.Component.Type.*;

@Component(
    name = "TimeStamperCustomMediationStepImpl",
    type = mediation,
    tooltip = "This custom code step adds the current time on the first
line of the message.",
    scope = "singleton",
    smallIconPath = "icons/TimeStamperCustomMediationStepImpl_16.png",
    largeIconPath = "icons/TimeStamperCustomMediationStepImpl_24.png"
)
```

您可以手动编辑此信息。

请注意，Studio 会将自定义元素的图标存储在项目的 src/icons 文件夹。有 16x16 像素和 24x24 像素 2 个版本。要使用自定义图标，请将 _16.png 和 _24.png 文件替换为您自己的图片。

显示自定义的程序集元素属性

如果您的自定义 Java 代码包含您希望由用户设置的属性，则可以在自定义元素的“Properties”视图的“Bean”选项卡上显示这些属性。

为此，请提供遵循 bean 模式的 getter 和 setter 方法。示例：属性 abc，提供 getter 和 setter 方法，例如 getAbc 与 setAbc。

然后添加 @Property 为以下任一对象添加注释 set 或 get 方法。该 name 值定义属性在“属性”视图中的标签，tooltip 该值定义当您将鼠标悬停在标签上时 Studio 显示的文本。您可以使用 order 值控制属性的

列出顺序。值较小的优先显示。一个 editelement value 指定在属性输入字段旁边显示的按钮类型，如下所示：

```
// Displays a text box with a Select button to enable you to select a Spring
bean.
@property(name="Text", edit=Edit.BEAN_REF)

// Displays a text box with a Select button to enable you to select a Java
class.
@property(name="Text", edit=Edit.CLASS)

// Displays a text box with a dialog box that enables you to specify a cron
expression.
@property(name="Text", edit=Edit.CRON)

// Displays a text box without a button.
@property(name="Text", edit=Edit.DEFAULT)

// Displays a Browse button to enable you to specify a file.
@property(name="Text", edit=Edit.FILE)

// Displays a Browse button to enable you to specify a folder.
@property(name="Text", edit=Edit.FOLDER)
```

Studio 会在包含正在编辑的程序集.xml 文件的项目的项目类路径中查找自定义类。要将自定义构成分发给多个程序集项目，请将二进制形式的自定义封装为包含类文件和图标图片的 JAR 文件。其他用户可在项目的类路径中包含此 JAR 文件，以便在其自己的 Assembly Editor 中使用该文件。

Assembly Editor 会在启动时自动加载在其类路径中找到的任何自定义构成。右键单击程序集图背景，并选择“Refresh Customizations”，以重新加载最近添加的所有自定义构成。

Spring Bean 配置

概念：Spring Bean 配置

您可以在 Assembly Editor 的“Properties”视图上的“Beans”选项卡中编辑 Spring Bean 属性。

Workday Studio 会在该视图左侧的顶层“beans”节点下显示程序集的所有 Bean。与“beans”节点关联的属性是全局属性，会影响程序集中的所有 Bean。

Bean 上方会显示各种选项。选项是否可用取决于您在 Bean 列表中所做的选择。选项包括：

选项	描述
Create 'bean'	在程序集文件中创建一个 Spring Bean 元素。要使此选项可用，请选择顶层“beans”节点。
Create 'alias'	为程序集文件中的 Spring Bean 元素创建别名。要使此选项可用，请选择顶层“beans”节点。
Create 'property'	为选定的 Bean 创建属性元素。
创建“constructor-arg”	创建 constructor-arg 所选 bean 的元素。
创建“ref”	创建 ref 元素，使其他 免责声明能够引用选定的属性。
创建“列表”	创建 list 元素来填充所选属性的属性，使您能够定义 List Java 集合的属性。

选项	描述
创建“地图”	创建 map 元素来填充所选属性的属性，使您能够定义 Map Java 集合的属性。
创建“导入”	创建最高层级 import 程序集的 es 元素，使您能够指定一个包含一个或多个 es 定义的资源文件。请选择顶层 Equity，使此选项可用。
创建“值”	创建嵌套 value 元素，这使您能够将属性指定为可读的字符串。

参考信息：全局 Bean 属性

属性	XML 属性名称	描述
默认自动关联	default-autowire	检查 BeanFactory 以自动解析免打扰。
默认自动配备候选人	default-autowire-candidate	使您能够根据与 bean 名称的模式匹配，将自动配备限制为一组候选项。
默认依赖关系检查	default-dependency-check	指定 Studio 是否检查未解析的 bean 依赖关系。
默认销毁方法	default-destroy-method	检查每个 bean 的指定销毁回调方法。
默认初始化方法	default-init-method	每个 bean 的初始化回调方法
默认延迟初始化	default-lazy-init	指定是在启动时还是在首次请求时创建 bean 实例。如果您不希望在使用 ApplicationContext，设置为 true。
默认合并	default-merge	确定 Studio 默认情况下是否合并集合。

参考信息：单个 Bean 属性

属性	XML 属性名称	描述
编号	id	程序集中自定义 元素的唯一编号。
摘要	abstract	指示 ibm 是否仅可用作父级 bean 定义 (将用作子级定义的模板)。默认值为 false。
自动关联	autowire	检查 BeanFactory 以自动解析关联要素。
自动配备候选人	autowire-candidate	使您能够根据与 bean 名称的模式匹配，将自动配备限制为一组候选项。
类	class	实施类名称

属性	XML 属性名称	描述
依赖关系检查	dependency-check	指示 Studio 是否会检查未解析的 bean 依赖关系。
取决于	depends-on	在初始化使用此元素的 ant 之前，强制初始化一个或多个 ant。要表示对多个 Bean 的依赖关系，请列出其名称，并用英文逗号、空格或英文分号分隔。
销毁方法	destroy-method	检查 bean 是否使用指定的销毁回调方法。
工厂 bean	factory-bean	当前容器或父级容器中的 bean，包含调用其实例方法会创建对象的实例方法。
工厂方法	factory-method	实例化 bean 时要使用的工厂方法的名称。
初始化方法	init-method	bean 的通用初始化方法。
延迟初始化	lazy-init	指示是应在启动时还是在首次请求时创建 bean 实例。如果您不希望在使用 ApplicationContext，设置为 true。默认情况下，此属性从容器级别设置中获取值。
名称	name	bean 的名称。
父级	parent	父级 元素。如果指定了此属性，则当前 Bean 将继承其配置数据。
主要	primary	将当前 bean 定义指定为主要自动关联候选项。默认值为 false。
范围	scope	从 bean 定义创建的对象的范围。默认值为 singleton。

参考信息：别名属性

属性	XML 属性名称	描述
别名	alias	别名。
名称	name	原始 bean 的编号。

参考信息：Property 属性

属性	XML 属性名称	描述
名称	name	属性名称。
java - java - java	spring-bean	容器中另一个 bean 的编号。
值	value	属性值。

参考信息：Constructor-Arg 属性

属性	XML 属性名称	描述
索引	index	构造函数参数的从零开始的索引。
java - java - java	spring-bean	容器中另一个 bean 的编号。在该 Bean 的定义中指定的构造函数参数将作为参数传递给所引用的 Bean 的构造函数。
类型	type	构造函数参数的类型。
值	value	构造函数参数的值。

参考信息：Ref 属性

属性	XML 属性名称	描述
bean	bean	通过指定 Name 或 Id 中的值之一来引用目标 bean。
Local	local	通过指定 Id 属性来引用本地目标 bean。允许 XML 解析器验证同一文件中的 XML 编号引用。
父级	parent	引用当前容器的父级容器中的目标 bean。在目标 Bean 的“Name”或“Id”中指定其中一个值。

参考信息：列表属性

属性	XML 属性名称	描述
合并	merge	指定是否合并 Java 列表集合。默认情况下，merge 属性从容器级别获取值。default-merge 设置
值类型	value-type	- Java 类型 value 列表中的元素。

参考信息：映射属性

属性	XML 属性名称	描述
合并	merge	指定是否合并 Java 映射集合。默认情况下，merge 属性从容器级别获取值。default-merge 设置
值类型	value-type	- Java 类型 value 列表中的元素。
密钥类型	key-type	- Java 类型 key 映射中的元素。

Message Builder

基于架构或 WSDL 创建 XML 字面量

开始之前

创建一个包含以下某个元素的程序集：

- “write”步骤
- “log”步骤
- “log-error”错误处理程序

关于此任务

通过 Workday Studio 的 Literal Builder，您可以创建符合所选架构的简单 XML 文档，或创建符合所选 WSDL 的 SOAP 请求消息。您可以为 SOAP 请求消息附加 WS-Security 标头。

过程

1. 向以下某个程序集元素添加“text”元素：
 - “write”步骤
 - “log”步骤
 - “log-error”错误处理程序
2. 在右侧窗格中，单击“Generate XML literal”。
3. 在“Literal Builder”向导中，选择要为其生成 XML 字面量的架构或 WSDL 元素。完成此任务时，请参考以下信息：

选项	描述
Show Messages	默认情况下，Studio 将 WSDL 元素显示为消息。要通过以下方式将这些维度显示为接口定义：portTypes、操作、输入消息和输出消息，请取消选择此选项。
换行 SOAP	将选定的 XML 定义封装在 SOAP 信封中。
添加 WS-Security 标头	添加 SOAP Header 元素添加到 SOAP 请求中。SOAP 标头包含您在 Web 服务中提供的 WS-Security 凭据。UsernameToken 元素。仅当您选择了“封装 SOAP”时可用。
Wrap XSLT	创建一个封装在 XSL 样式表中的 XML 字面量。您可将样式表另存为 XSLT 文件以供日后使用。
Ask for Choices Selection	在您创建和复制 XML 字面量或 SOAP 字面量时，提示您从架构中选择特定选项。如果您不选择此选项，Workday Studio 将使用特定集中的第一个选项。

结果

Workday Studio 会显示 XML 字面量的预览，您可以对其进行编辑。

基于架构或 WSDL 创建 XPath 表达式

开始之前

创建一个包含以下某个步骤的程序集：

- java-to-xml
- validate
- xml-to-java

关于此任务

在 Workday Studio 中，您可使用 XPath Expression Builder，基于所选架构或 WSDL 创建 XPath 表达式。

过程

1. 在以下某个步骤的“Properties”视图中，单击“XPath”属性旁边的“Build XPath Expression”：
 - java-to-xml
 - validate
 - xml-to-java
2. 在“XPath Expression Builder”向导的下部窗格中，选择 XPath 表达式将与之匹配的架构或 WSDL 元素。完成此任务时，请参考以下信息：

选项	描述
Show Messages	默认情况下，Studio 将 WSDL 元素显示为消息。要通过以下方式将这些维度显示为接口定义：portTypes、操作、输入消息和输出消息，请取消选择此选项。
前缀 SOAP	默认情况下，Workday 会使用以下表达式为您创建的 XPath 表达式添加前缀，如上方的文本窗格中所示：env:Envelope/env:Body。要移除前缀，请取消选择此选项。

3. 如有需要，您可在上部文本窗格中手动编辑 XPath 表达式。示例：您可能希望使用谓词来筛选 XPath 返回的节点集。

概念：Workday Studio 消息构建器

Workday Studio 的“Message Builder”提供了一个图形视图，可简化以下程序集元素的消息内容定义：

- “write”步骤
- “log”步骤
- “log-error”错误处理程序

Assembly Editor 中的“Message Builder”选项卡提供了一个集中位置，供您为程序集中的任何相关元素定义消息内容。“Message Builder”也以选项卡的形式出现在各个“write”“log”“log-error”元素的“Properties”视图中。

通过左侧的“Message Builder”窗格，您可以定义消息内容类型。使用顶部的选项可添加各种类型的消息元素：

选项	描述
Create 'message-content'	将中介消息的内容添加到消息中。
Create 'text'	为消息添加字符串字面值。有以下 3 个选项： • 手动输入文本，然后单击“Apply changes”。

选项	描述
	- 单击“Insert MVEL Expression”，输入 MVEL 表达式（如有必要，可与纯文本组合使用），然后单击“Apply”。 - 单击“Generate XML literal”，然后选择定义模式或 WSDL 元素。
Create 'exception'	如果中介上下文中存在异常，则向消息添加异常。仅用于“log-error”错误处理程序。
Create 'line-separator'	为消息添加换行符。
Create 'xpath'	将 XPath 表达式的结果添加到消息中。
Create 'xslt'	将 XSL 转换的结果添加到消息中。
Create 'xml-stream'	创建“log”或“write”步骤的内容。
Create 'regex-match'	向消息内容应用正则表达式，并将所有匹配的实例添加到中介消息。
Create 'static-file'	指定一个输入文件，并使用其全部内容填充消息。
Create 'message-store-id'	添加消息的消息存储库编号。仅在以下情况下适用：store-message 传输属性。
创建“表头”	为消息添加指定消息标头的值。
Create 'headers'	以“名称-值”标头对的形式添加完整的标头集，并用换行符分隔。
创建“customer-id”	从“”中添加客户编号 MediationContext 添加到消息。如果未设置客户编号，则在消息中显示“null”。

您可在右侧窗格中定义消息元素的值。值窗格的内容因消息元素的类型而异。

要对消息元素进行重新排序或删除，请使用内容窗格右侧的按钮。它们在“Message Builder”中的顺序会体现在 XML 中。

参考信息：消息元素属性

使用“Message Builder”顶部的按钮，将消息元素添加到“write”“log”“log-error”元素中。

并非每个消息元素都具有属性。

message-content

属性	XML 属性名称	描述
限制	limit	对消息中包含的字符数进行了限制。默认值为零，表示无字符数限制。

exception

属性	XML 属性名称	描述
转义 Xml	escape-xml	允许在 写入 步骤中出现异常时转义 XML。

xpath

属性	XML 属性名称	描述
命名空间	namespaces	让您能够采用以下格式指定 XPath 表达式所需的任何命名空间 <prefix1> <namespace1> <prefix2> <namespace2>。
XPath	xpath	用于从传入消息中提取值的 XPath 表达式。
XPath 版本	xpath-version	Workday Studio 支持 XPath 版本 1.0 和 2.0。对于 Workday 12 及更高版本的程序集，默认值为 2；对于较早版本，默认值为 1。

xslt

属性	XML 属性名称	描述
Allow Saxon Bytecode Generation		指定是否允许生成 Saxon 字节码。默认值为 false。
重用策略		指定 XSLT 转换器重用策略： - 默认值：如果您指定 url 属性（使用 mctx URL 协议或 MVEL 表达式），则默认行为为 no-reuse。否则为 transformer-pool。 - 注：当您将 Transformer-Factory 设置为“Saxon”时，默认行为为 templates。 - no-reuse：Workday 不会重复使用转换器对象。 - templates：Workday 会构建一个 Templates 对象，并据其新建转换器对象。这些转换器对象不会被重复使用。 - transformer-pool：Workday 使用 Transformer 池。如果转换器对象缓存了值，则不建议使用此项。
Secure Processing		指定转换器是否仅执行 XSLT 标准处理。若设置为“true”，系统将对 XML 构造施加限制。示例：忽略 JavaScript 或 Java 扩展处理。若设置为“false”，则可根据 XML 规范处理 XML。
Workday Transformer		的类名称 TransformerFactory 您希

属性	XML 属性名称	描述
		望使用的类型默认值为 JAXP 工厂。您还有其他 3 个选项： • <code>net.sf.saxon.TransformerFactory</code> (对于 撒克语) • <code>org.apache.xalan.processor.Tr</code> 适用于 Xalan。 • <code>net.sf.joost.trax.TransformerF</code> 对于 Joost。
URL		XSLT 文件位置 (相对路径或绝对 URL)。支持 MVEL 表达式模板。 您可以使用 <code>mctx</code> (中介上下文) URL 协议直接从消息内容加载 XSLT。示例：要加载附件中包含的 XSLT 文件，请使用 <code>mctx:parts/1</code>。

xml-stream

属性	XML 属性名称	描述
命名空间	namespaces	让您能够采用以下格式指定 XPath 表达式所需的任何命名空间 <code><prefix1> <namespace1> <prefix2> <namespace2></code> 。
XPath	xpath	检索 XML 元素时应用于消息的 XPath 表达式。Workday 使用 Streaming API for XML (StAX) 处理。此库会读取 XML 消息并将其转换为事件，而非 DOM。它本身不执行任何转换。 该 XPath 表达式与您在 <code>xml-stream-splitter</code> 策略。Workday 支持限定的或本地化的元素名称和星号。不支持完整的 XPath 标准。 示例： • <code>/order/item, /ns:order/ns:order</code> • <code>//item, //ns:item, /ns:order/item</code> • <code>/*/*, /** /*/*</code>
XPath 版本	xpath-version	Workday Studio 支持 XPath 版本 1.0 和 2.0。对于 Workday 12 及更高版本的程序集，默认值为 2；对于较早版本，默认值为 1。
限制	limit	根据 XPath 表达式限制要匹配的 XML 片段的数量。主要出于

属性	XML 属性名称	描述
		性能原因有用。如果您知道只有 1 个匹配的 XML 片段，请将限制设置为 1 以防止在检索结果后对文档进行拉取解析。默认值为 -1，表示结果集不受限制。

regex-match

属性	XML 属性名称	描述
分隔符	delimiter	指定一个字符串，用于分隔 2 个或更多匹配项。默认值为一个空格。
限制	limit	指定要在结果集中包括的事件数。默认值为 0，表示可以包含所有实例。
正则表达式	regex	定义匹配表达式。

static-file

属性	XML 属性名称	描述
字符编码	character-encoding	指定字符编码。默认值为 UTF-8。
输入文件	input-file	文件。

header

属性	XML 属性名称	描述
包括名称	include-name	指定 Studio 是否插入标头名称。
名称	name	表头名称。
未设置值	not-set-value	在标头未找到或为空时发出的字符串。

传输绑定

概念：传输绑定

传输绑定是可选的子元素，您可在 Workday Studio 的 out-transport 中进行配置。通过传输绑定，out-transport 能够支持指定格式的 Web 服务。

示例：如果您想调用 RESTful Web 服务，可以在“http-out”传输中配置“rest-binding”子元素。

您可在以下 out-transport 中配置传输绑定：

- http-out
- ftp-out
- ftps-out
- sftp-out
- xmpp-out

- custom-out
- email-out

Workday Studio 支持以下传输绑定：

传输绑定	描述	注释
rest-binding	使您能够使用 HTTP 请求方法调用 RESTful Web 服务。	支持以下 HTTP 方法： • GET • PUT • POST
wsdl-http-binding	使您能够使用 HTTP 请求方法和 WSDL 文件来调用 Web 服务。	支持以下 HTTP 方法： • GET • PUT • POST
wsdl-soap-binding	使您能够使用 HTTP 请求方法调用基于 SOAP 的 Web 服务。	支持以下 HTTP 方法： • GET • PUT • POST 支持以下子元素： • custom-policy • ws-rm • ws-security • ws-ut

参考信息：WSDL-HTTP-Binding 属性

“Common”选项卡

属性A	XML 属性名称	描述
端点名称	endpoint-name	wsdl-http-bounding 元素访问 Web 服务的端点。
接口名称	interface-name	该 portType Web 服务的 (或接口) 名称。
服务名称	service-name	Web 服务的名称。
WSDL 文件	wsdl-name	描述 Web 服务的 WSDL 文件。
将参数序列化	serializeParams	当前未使用。

参考信息：WSDL-SOAP-Binding 属性

“Common”选项卡

属性A	XML 属性名称	描述
端点名称	endpoint-name	wsdl-soap-bounding 元素访问 Web 服务的端点。

属性A	XML 属性名称	描述
接口名称	interface-name	该 portType Web 服务的 (或接口) 名称。
服务名称	service-name	Web 服务的名称。
WSDL 文件	wSDL-name	描述 Web 服务的 WSDL 文件。
故障处理	fault-handling	指定 wsdl-soap-bounding 元素对 HTTP 500 响应代码的错误处理流程。选项包括： • ignore • error • abort

WSDL-SOAP-Binding 子元素

概念：WSDL-SOAP-Binding 子元素

“wsdl-soap-Binding”传输绑定支持以下子元素：

子元素	描述
custom-policy	对于 out-transport，支持自定义 Web 服务安全策略。
ws-rm	对于 out-transport，支持消息传递安全策略。
ws-security	在 out-transport 上实施一个本地文件，该文件包含 SOAP 请求的 Web 服务安全策略。
ws-ut	使 Workday 集成服务器能够在程序集的 out-transport 中提取、验证和传播用户名及密码安全凭据。

参考信息：Custom-Policy 属性

“Common”选项卡

属性A	XML 属性名称	描述
方法	method	对您的自定义安全策略调用的方法。
java - bean	ref	实施您的自定义安全策略的自定义 Spingeer。

参考信息：WS-Rm 属性

“Common”选项卡

属性A	XML 属性名称	描述
文件	file	描述您的消息传递安全策略配置的文件的路径。

参考信息：WS-Security 属性

“Common”选项卡

属性A	XML 属性名称	描述
文件	file	描述 Web 服务安全配置的文件的路径。

参考信息：WS-Ut 属性

“Common”选项卡

属性A	XML 属性名称	描述
用户名	username	将用户名插入到 () 中的出站 SOAP 消息的安全标头。MediationContext。
密码	password	将密码插入到出站 SOAP 消息的安全标头。MediationContext。
密码摘要算法	password-digest-algorithm	对中的密码应用哈希算法，这些密码包含在出站 SOAP 消息的安全标头中。MediationContext。选项包括： • SHA-1 • SHA-256 • SHA-384 • SHA-512 • none
java - java - java	ref	一个可选的 Spingle bean，用于对中的出站 SOAP 消息的安全标头中的用户名和密码值进行身份验证。MediationContext。
已创建	created	在 中的出站 SOAP 消息的安全标头中创建时间戳元素。MediationContext。
随机数	nonce	在 中的出站 SOAP 消息的安全标头中创建一个随机数元素。MediationContext。

Schema Explorer

显示 WSDL 和 XSD 文件

关于此任务

“Schema Explorer”视图以只读格式显示 WSDL 和 XSD 文件。您可以将这些文件中的内容复制到剪贴板，从而在 Studio 的其他文本编辑器视图中重复使用这些数据。

过程

- 1. 选择“窗口” > “显示视图” > “模式浏览器”。
- 2. 在“模式浏览器”视图中，单击“添加 WSDL 或 XSD”。
- 3. 选择一个 WSDL 或 XSD 文件。
- 4. 在“Schema Explorer”视图中，展开该 WSDL 或 XSD 文件以查看其结构。

结果

Workday Studio 会在“Schema Explorer”视图中显示该 WSDL 或 XSD 文件。

下一步做什么

将该 WSDL 或 XSD 文件中的内容复制到剪贴板。

显示 Workday Web 服务 WSDL

关于此任务

“Schema Explorer”视图以只读格式显示 WSDL 和 XSD 文件。您可以将这些文件中的内容复制到剪贴板，从而在 Studio 的其他文本编辑器视图中重复使用这些数据。

过程

- 1. 选择“窗口” > “显示视图” > “模式浏览器”。
- 2. 在“架构浏览器”视图中，单击“添加 Workday Web 服务”。
- 3. 选择并展开一项 Workday Web 服务。
- 4. 选择“wsdl”。
- 5. 在“Schema Explorer”视图中，展开该 Workday Web 服务 WSDL 以查看其结构。

结果

Workday Studio 会在“Schema Explorer”视图中显示该 Workday Web 服务 WSDL。

下一步做什么

将内容从 Workday Web 服务 WSDL 复制到剪贴板。

显示自定义 Raas 报告架构

关于此任务

“Schema Explorer”视图以只读格式显示 WSDL 和架构 (XSD) 文件。您可以将这些文件中的内容复制到剪贴板，从而在 Studio 的其他文本编辑器视图中重复使用这些数据。

过程

- 1. 选择“窗口” > “显示视图” > “模式浏览器”。
- 2. 在“模式浏览器”视图中，单击“添加自定义 Raas 报告模式”。
- 3. 在“Add Custom Raas Report”窗口中，请参考以下信息：

选项	描述
环境	包含您的自定义报告的 Workday 环境。
ReportName (报告名称)	自定义报告的名称。

- 4. 在“Select Reports”窗口中，选择一个自定义报告。

5. 单击“Finish”。
6. 在“Schema Explorer”视图中，展开自定义报告架构以查看其结构。

结果

Workday Studio 会在“Schema Explorer”视图中显示自定义报告架构。

下一步做什么

将自定义报告架构中的内容复制到剪贴板。

概念：Schema Explorer

“Schema Explorer”视图以只读格式显示 WSDL 和 XSD 文件。您可以将这些文件中的内容复制到剪贴板，以便在 Workday Studio 的其他文本编辑器视图中重复使用这些数据。

您可以在“type filter text”字段中输入关键字，以筛选“Schema Explorer”视图的内容。

默认情况下，“Schema Explorer”视图显示在屏幕的右上方。您可以将“Schema Explorer”视图拖放到 Workday Studio 用户界面上您所偏好的位置。

MVEL

概念：MVEL 基础知识

语法

MVEL 是一种表达式语言，适用于基于 Java 的应用程序。MVEL 的语法类似于 Java，但增加了一些内容，这使得它在内容匹配方面更高效。

注意：

- MVEL 使用 Java 命名空间和类，但它本身不能声明两者。
- 与 Java 不同，MVEL 是动态类型的语言，并且类型是可选的。
- MVEL 表达式可以是单个语句，也可以是多行脚本。对于多行脚本，请使用分号分隔各语句。
- MVEL 根据“最后一个值转出”原则运行。它支持 return 关键字，但通常不需要它。
- MVEL 支持运算符优先级规则，并使用括号来控制执行顺序。
- 您可以使用单引号或双引号将 MVEL 字符串文本括起来。
- 在 MVEL 中，分号是保留字符。不能在表达式中使用。用途 (char) 59 改为。示例：
props['string']='a;b' 引发错误。此代码不会：

```
props['string']= (char) 59
props['string_new']='a'props['string']'b'
```

MVEL 表达式可包括：

- 布尔值表达式
- 方法调用
- 属性表达式
- 变量指定

以下是简单的 MVEL 属性表达式示例：

```
context.errorMessage
```

它会提取属性 errorMessage 从变量或上下文对象 context。

MVEL 支持所有常用的 Java 比较运算符、数学运算符和逻辑运算符。下表汇总了其他重要的 MVEL 运算符：

运算符	描述	示例
<code>new</code>	实例化一个对象。	<code>new String("foo")</code>
<code>with</code>	对单个对象实例执行多项操作。	<code>with (value) { name = 'Foo', age = 50, sex = Sex.MALE }</code>
<code>assert</code>	对某个值是否为 <code>true</code> 进行断言，否则失败，并返回 <code>AssertionError</code> 。	<code>assert foo != null</code>
<code>contains</code>	确定左侧的值是否包含右侧的值。	<code>var contains "Foo"</code>
<code>is 或 instance of</code>	确定左侧的值是否为右侧类的成员。	<code>var instanceof String</code>
<code>strsim</code>	比较字符串，并返回相似度（以百分比表示）。	<code>"foobol" strsim "foobar"</code>
<code>soundslike</code>	在 2 个字符串之间执行 <code>soundex</code> 比较。	<code>"foobar" soundslike "fubar"</code>
<code>&</code>	执行“按位与”运算。	<code>foo & 5</code>
<code> </code>	执行“按位或”运算。	<code>foo 5</code>
<code>^</code>	执行按位异或运算。	<code>foo ^ 5</code>
<code>+</code>	用于连接 2 个字符串的重载运算符。	<code>"foo" + "bar"</code>
<code>#</code>	将 2 个文本连接为字符串。	<code>8 # 9</code>
<code>in</code>	检查集合内的对象。	<code>(foo in list)</code>
<code>=</code>	为变量指定值。	<code>var = "foobar"</code>

值测试

在 Java 中，运算符 `==` 检查引用是否指向同一个对象。但是，在 MVEL 中，比较运算符 `==` 检查值是否相等。MVEL 表达式 `foo == 'bar'` 与 Java 的等效项 `foo.equals("bar")`。

MVEL 布尔值表达式的示例：

```
user.name == 'John Smith'
```

它会返回 `true` 如果 `name` 变量或上下文对象的属性 `user` 是 John Smith。

MVEL 采用类型强制。如果您比较 2 个不同类型的值，MVEL 会尝试将右侧值强制转换为左侧的类型。如果无法执行此操作，则会尝试将左侧值强制转换为右侧的类型。示例：

```
"123" == 123;
```

此表达式将返回 `true` 因为 MVEL 在执行比较之前会将右侧的值强制转换为字符串。

列表、映射和数组

将 MVEL 列表括在方括号中。使用逗号分隔各个条目。示例：

```
["Paris", "London", "Berlin"]
```

将 MVEL 映射用方括号括起来。用冒号分隔键和值。示例：

```
["Foo" : "Bar", "Bar" : "Foo"]
```

将 MVEL 数组括在大括号内。使用逗号分隔各个条目。示例：

```
{"Bill", "Bob", "Ben"}
```

属性导航

MVEL 为访问属性、静态字段、列表、映射和数组提供了单一、统一的语法。

在 Java 中，您可以使用以下语句从对象访问属性：

```
context.getException().getCause()
```

在 MVEL 中，您可以使用以下表达式访问同一属性：

```
context.exception.cause
```

以相同的方式访问列表和数组。示例：

```
parts[2]
```

您可以像访问数组一样访问映射，但可以将任何对象作为索引值进行传递。示例：

```
props["foobar"]
```

如果映射使用字符串作为键，您可以将映射本身视为虚拟对象。示例：

```
props.foobar
```

出于索引目的，MVEL 将字符串视为数组。

此示例将返回 B：

```
foo = "Berlin";  
foo[0];
```

指定

您可以在 MVEL 表达式中指定变量，以在表达式中使用或在运行时提取。

MVEL 是一种动态类型的语言，因此您在声明新变量时不需要指定类型，但您可以指定。

以下两个示例在 MVEL 中都有效：

```
str = "Some string";  
String str = "Some string";
```

为类型化变量指定值时，MVEL 会尝试执行自动类型转换。示例：

```
String num = 1;
```

您可以将动态类型化变量的值转换为另一种类型。示例：

```
String num = 1;
```

版块：键入和强制

MVEL 使用动态类型，并提供了可选的静态类型。原生 Java 对象为静态类型。为与这些对象进行交互，MVEL 有时必须执行强制类型转换，这可能会对性能产生影响。

MVEL 会自动将类型强制转换为最适合当前值的类型。这意味着您在处理数字时必须小心。浮点运算中可能会出现错误，因为 MVEL 会自动放弃无限精度类型转换。java.math.BigDecimal 至 java.lang.Double 与 java.lang.Long。当存储值的类型不是您预期的类型时，这可能会导致问题。通常，您应避免对 MVEL 中的财务值执行非整数算术运算。

使用 MVEL 的强制类型转换意味着，您可以在不调整输入的情况下调用 Java 方法。解释器或编译器会分析您要传递给方法的类型，并确定所需的强制类型转换。

如果发生数组类型冲突，MVEL 会尝试将整个数组强制转换为所需输入类型。

如果所赋的值可以强制转换，MVEL 会对静态类型变量执行任何必要的强制转换。如果不可以，则会发生异常。

概念：MVEL 与 Workday Studio

在 Workday Studio 中，MVEL 可让您检查和操作程序集中的消息部分与构成参数。您可以使用 MVEL 执行以下操作：

- 从消息中提取信息。
- 设置数字和日期格式。
- 替换字符串中的特定字符。
- 检索集成属性或启动参数。

您应避免使用 MVEL 来操作较大的数据字符串。示例：将消息转换为字符串可能会产生性能和可扩展性问题。

在支持 MVEL 的字段中，您可以同时按 Ctrl 和空格键来显示内容助手弹出窗口。

MVEL 表达式可以是单个语句，也可以是多行脚本。对于多行脚本，请使用分号分隔各行。

注：Workday Studio 将多行表达式作为一个整体进行求值，而不是按顺序求值。

支持 MVEL 的字段会展开，以容纳新行。行数达到 6 行后，这些字段会停止展开，并变为可滚动状态。Workday Studio 会对 MVEL 表达式应用语法着色。

MVEL 在执行期间会实施运算符优先级规则。您可以使用方括号来控制 MVEL 表达式的执行顺序。

注：由于是在内存中处理，因此 XPath 方法存在 100 万个字符的限制，例如 xpath，xpathB 和 xpathF。即使查询超出此限制，您也不会收到错误消息。您应使用流式 API，例如 写入 步骤的 xml-stream 元素（如果您需要处理大型消息）。

参考信息：Workday Studio 帮助程序对象

Workday Studio 提供了许多帮助程序对象，使您能够使用 MVEL 访问集成的运行时属性。您可以使用帮助程序对象对运行时值进行简单更改，或者提取这些值并将其传递给程序集中的构成。

帮助程序对象	目的
context	让用户有权访问 MediationContext object - 即集成在运行时的动态状态。该

帮助程序对象	目的
	MediationContext 对象链接到当前消息、中介属性、中介变量和当前错误信息。 示例：此表达式将检索属性的值 myProperty 从 MediationContext： <pre>context.getProperty('myProperty')</pre> 示例：此表达式将检索属性 someProperty 从 MediationContext 如果属性值设置为 someValue： <pre>context.getProperty('someProperty').equals('someValue')</pre>
da	“Document Accessor”辅助对象。支持访问在此集成运行之前附加到集成事件的文档列表。可有效地将“Get Event Documents”Web 服务调用与对 Web 服务响应中所引用文档的检索相结合。 该 da 辅助对象具有一些限制： • 它必须明确包含逻辑，用于处理可用附件数量超出预期的情况。 • 它无法访问附件的文件类型。 • 它无法访问附件的字符编码。 “doc-iterator”路由策略会自动处理数量超出预期的附件，并且可以访问多用途互联网邮件扩展（MIME 类型）和字符编码。对于集成而言，使用该策略（而非此帮助器对象）来检索输入文档通常更为合适。 示例：以下表达式从文件名为“eib_transform_output.txt”的集成事件中检索文档，并添加名为“”的变量。var1： <pre>da.toVar('eib_transform_output.txt', 'var1')</pre> 示例：以下表达式将检索具有给定标签的文档： <pre>da.getDocsMatching('label')</pre>
ftp	让用户有权访问向 ftp、ftps 或 sftp 服务器发出 LIST 请求时返回的文件列表。 示例：以下表达式可检索 URL 对应的文件列表： <pre>ftp.list('URL')</pre>
intsys	让用户能够便捷且高效地访问“Get IntegrationSystems”Web 服务调用所返回的集成系统配置。 示例：以下表达式将检索名为“”的集成属性。username： <pre>intsys.getAttribute('username')</pre>

帮助程序对象	目的
	示例：如果存在具有指定名称的集成服务，则以下表达式将显示为 true： <pre>intsys.isServiceEnabled('name')</pre> 注：如果您使用 intsys 要访问属性，您必须先在上定义服务 workday-in 配置文件（用于集成系统的传输）。
lp	让用户有权访问集成启动事件中的启动参数和其他信息。 示例：以下表达式将检索具有给定名称的启动参数的值： <pre>lp.getSimpleData('name')</pre> 示例：如果存在具有给定标签的参数，则以下表达式显示为 true： <pre>lp.exists('label') == true</pre> 注：在使用之前 lp 变量，则需要将启动参数 XML 指定给该变量 wd.launchparameters。
message	提供对当前消息及其标头信息的访问权限。使用此对象可为消息指定新内容。 示例：此表达式从“”中以文本形式检索消息的根部分。MediationMessage 接口： <pre>message.rootPartAsText</pre> 示例：此表达式会检索 MediationContext 变量： <pre>message.variables</pre>
mtable	让用户有权访问 MTableAdapter 该接口定义了对某个 mtable 实例。 示例：以下表达式用于设置“”中的数据 mtable： <pre>mtable['my_property'].set(0, 'Data_Column', 'my_test_data')</pre>
parts	提供对当前消息各个部分的访问权限。Workday 集成中的大多数消息都有一个部分，即根部分。用途 parts[0] 以访问根部分。 示例：此表达式将检索标头 someHeader 从根部件： <pre>parts[0].getHeader('someHeader')</pre>

帮助程序对象	目的
	示例：此表达式实现 xpathB 方法来评估 XPath 布尔值表达式： <pre>parts[0].xpathB('/a/b')</pre>
props	让用户有权访问 MediationContext 属性。 示例：此表达式将检索属性的值 myProperty 从 MediationContext： <pre>props.myProperty</pre> 示例：此表达式将检索属性的值 bean.property 从 MediationContext： <pre>props['bean.property']</pre>
spring	让用户有权访问在程序集的 Spingle 配置中定义的创建 bean 和检索 bean。 示例：此表达式会检索 name 名为“”的 bean 的属性 Fred 在春季上下文中： <pre>spring.getBean('fred').name</pre> 示例：以下表达式可访问春季应用程序上下文： <pre>spring.parent ApplicationContext</pre>
util	一组实用函数，用于执行常见操作，例如对值进行转义以插入 XML 中，或创建 HTTPS URL。 示例：此表达式会检查当前消息是否在 MediationContext 是拆分器组件将生成的最后一条消息： <pre>props['is.last.record'] = util.isLastMessageInBatch()</pre> 部分 util 函数具有严重的限制。示例：的 xpathToCommaDelimString 函数根据对给定 XML 字符串进行运算的 XPath 表达式所返回的节点序列的字符串值，生成一个逗号分隔列表。然后，您可以使用 MVEL 中的 Java 字符串操作函数对逗号分隔列表进行拆分和操作。但是，该函数具有以下限制： • 它仅支持 XPath 1。 • 它无法生成 CSV 文件，因为它无法正确转义换行符、逗号和引号字符。 • 它无法可靠地分隔包含逗号的数据。 • 它要求使用字符串形式的 XML，这会促使用户使用不可扩展的做法，例如将当前上下文消息或变量转换为字符串。 注：出于稳定性和向后兼容性考虑，Workday 强烈建议使用受版本控制的显式 Web 服务请求。使用

帮助程序对象	目的
	MVELUtilHelper.getLatestWWSVersion() 在集成中使用""不是最佳做法，并且可能会导致集成失败或出现意外结果。
vars	让用户有权访问 MediationContext 变量。 示例：以下表达式将检索变量的长度 myVariable： <pre>vars['myVariable'].length</pre> 示例：以下表达式将检索变量的长度 myVariable： <pre>vars['myVariable'].length</pre>
xmldiff	让用户以编程方式访问使用 xmldiff 组件进行文档比较的结果。 示例：以下表达式检查 2 个文档之间是否存在差异： <pre>xmldiff.isDifferent() == true</pre> 示例：以下表达式会检索 2 个文档之间的差异列表： <pre>xmldiff.changes</pre>

参考信息：文档访问器帮助程序对象

实用函数

名称	参数	返回值类型	注释
getData	整数索引	来源	在指定的从零开始的索引处检索文档。结合使用 message.setMessage 帮助器函数，以节省内存的方式使用检索到的数据填充当前消息。 示例：以下代码使用从第一个附件中检索到的 UTF-8 XML 消息填充当前消息： <pre>message.setMessage(da.getDa xml')</pre> 如果没有具有指定索引的集成附件，将导致错误。

名称	参数	返回值类型	注释
	字符串文件名	来源	检索具有指定文件名的文档。结合使用 <code>message.setMessage</code> 帮助器函数，以节省内存的方式使用检索到的数据填充当前消息。 示例：以下代码使用名为“input.xml”的 UTF-8 XML 消息附件填充当前消息： <pre>message.setMessage(da.getData('input.xml'))</pre> 如果没有具有指定文件名的集成附件，将导致错误。
<code>getDocumentIndexes</code>	列表 <String> 标签	列表<整数>	返回一个 Java 整数列表，其中列出了以所有指定标签来标记的文档的索引。 请注意，如果其中一个标签为“*”，则任何标签都会匹配，并且此函数会返回至少具有 1 个标签的文档的所有索引。
<code>getDocsMatching</code>	列表<String>	列表<源>	检索以所有指定的标签来标记的文档。 注: Workday 会将小于 100 KB 的文档存储在内存中，并将较大的文档存储在磁盘上。请注意运行时内存和磁盘空间限制。
	字符串标签	来源	检索以指定的标签来标记的文档。 注: 此处的标签是指集成附件标签。与文档标记同义。
	字符串[] 标签	来源	在字符串数组中检索标有所有标签的文档。
<code>getFileName</code>	整数索引	字符串	检索具有指定索引的文件名。
	字符串标签	列表<字符串>	检索以指定标签来标记的所有文档的文件名 Java 列表。

名称	参数	返回值类型	注释
	列表<字符串> 标签	列表<字符串>	检索以指定标签来标记的所有文档的文件名 Java 列表。
	字符串标签、整数索引	字符串	检索使用指定标签标记的所有文档中的第 n 个文件名。 在功能上等价于以下代码： <pre>da.getFileNames(label).get(</pre> 在大多数情况下，您必须使用 <pre>da.getFileNames(label)</pre> 要先查找匹配文件名的数量，然后才能使用索引。唯一的例外是极少数情况，即您需要第一个值（零索引）。
	列表<字符串> 标签、整数索引	字符串	检索使用指定标签标记的所有文档中的第 n 个文件名。 在功能上等价于以下代码： <pre>da.getFileNames(labels).get</pre> 在大多数情况下，您必须使用 <pre>da.getFileNames(labels)</pre> 要先查找匹配文件名的数量，然后才能使用索引。唯一的例外是极少数情况，即您需要第一个值（零索引）。
	字符串[] 标签、整数索引		检索使用指定标签标记的所有文档中的第 n 个文件名。 在功能上等价于以下代码： <pre>da.getFileNames(java.util.A</pre> 在大多数情况下，您必须使用 <pre>da.getFileNames(labels)</pre> 要先查找匹配文件名的数量，然后才能使用索引。唯一的例外是极少数情况，即您需要第一个值（零索引）。

名称	参数	返回值类型	注释
hasFile	字符串文件名	布尔值	确定是否存在具有指定文件名的集成附件。
	字符串标签、字符串文件名	布尔值	确定是否存在具有指定文件名并以指定标签来标记的集成附件。
	列表<字符串> 标签、字符串文件名	布尔值	确定是否存在具有指定文件名且标记有列表中所有标签的集成附件。
sort	字符串顺序	无	按文件名对附加文档进行排序。有效顺序值为： "FILENAME_ASCENDING" "FILENAME_DESCENDING" 任何不包含文件名的附件都放在最后面。
size	无	整数	返回附加文档的数量。
	列表<字符串> 标签	整数	返回以所有指定标签来标记的文档的数量。
	字符串标签	整数	返回以指定标签来标记的文档的数量。
	字符串[] 标签	整数	在字符串数组中返回标有所有标签的文档数。
toVar	整数索引、字符串 var_name	MessageVariable	检索指定索引处的文档，并以高效使用内存的方式将该文档放入指定的程序集变量中。 示例： <pre>da.toVar(0, 'myInputDocument')</pre>
	String filename, String var_name	MessageVariable	检索具有指定文件名的文档，并以高效使用内存的方式将该文档放入指定的程序集变量中。 示例： <pre>da.toVar('WorkerData.xml', 'm')</pre>

参考信息：启动参数帮助程序对象

实用属性

名称	注释
integrationEventWID	返回与当前集成启动相关的集成事件的 Workday 编号。在进行 Web 服务调

名称	注释
	用 (例如 <code>Get_Event_Documents</code> , <code>Put_Integration_Message</code> , 或 <code>Put_Integration_Event</code> 。 自动向 <code>wcc://GetEventDocuments</code> 、 <code>wcc://PutIntegrationMessage</code> 和 <code>wcc://PutIntegrationEvent</code> 公共组件提供默认参数。
<code>integrationSystem</code>	返回 <code>Integration_System_ID</code> 当前集成系统中的“”。在为事件订阅或字段重设请求填充 Web 服务调用时，此选项很有用。
<code>integrationSystemRefWID</code>	返回当前集成系统的 Workday 编号。完全等效的替代方案 <code>integrationSystem</code> 属性。
<code>parentEventWID</code>	返回当前启动事件的父级事件的 Workday 编号。
<code>sentOn</code>	返回集成的启动日期和时间，格式为 ISO-8601 日期和时间。
<code>sentOnAsDate</code>	以以下格式返回集成的启动日期和时间： <code>java.util.Date</code> 对象。

实用的函数

注:

其中一些函数仅通过启动参数的名称来指定启动参数。其他函数则同时指定启动参数的名称和提供该参数的集成服务的名称。如果集成系统上配置的多个集成服务中存在同名的启动参数，则后者很有用。大多数集成无需应对在多个服务中重复的启动参数名称，因此不常用到这些函数。

名称	参数	返回值类型	注释
<code>exists</code>	字符串名称	布尔值	如果指定的启动参数具有非空值，则返回 true。
	字符串 <code>service_name</code> 、 字符串名称	布尔值	如果指定集成服务中的指定启动参数具有非空值，则返回 true。 仅当集成系统上启用的多个集成服务中存在同名的启动参数时，才优先使用单参数版本。
<code>getDate</code>	字符串名称	字符串	返回运行时参数的字符串格式日期值。 格式为“YYYY-MM-DDZ”或“YYYY-MM-DD +/-偏移”。
	字符串 <code>service_name</code> 、 字符串名称	字符串	返回指定服务的运行时参数的字符串格式日期值。格式为 YYYY-MM-DDZ 或 YYYY-MM-DD +/- 偏移。
<code>getParameterData</code>	字符串名称	<code>org.w3c.dom.Element</code>	返回 XML 元素。 <code>Integration_Runtime_Parameters</code>

名称	参数	返回值类型	注释
			中包含来自的已命名启动参数 Launch_Integration_Event 启动时发送到集成的消息。
	String service_name, String 名称	org.w3c.dom.Element	返回 XML 元素。 Integration_Runtime_Parameter 中包含中的指定服务的已命名启动参数 Launch_Integration_Event 启动时发送到集成的消息。
getReferenceData	字符串名称、字符串类型	字符串	根据具有指定名称的启动参数，返回给定类型的参考编号。 示例：以下代码根据名为“Employee”的启动参数返回员工编号： <pre>lp.getReferenceData('Employee')</pre>
getReferenceDataList	字符串名称、字符串类型	列表<字符串>	根据多实例启动参数，返回一个包含零个或多个参考编号的 Java 列表。
	字符串 service_name、 字符串名称、字符串类型	列表<字符串>	返回一个 Java 列表，其中包含指定服务的多实例启动参数中的零个或多个参考编号。
getSequencedValue	字符串名称	字符串	返回具有指定名称的序列生成器的当前值。
	字符串 service_name、 字符串名称	字符串	返回在指定服务中声明的指定序列生成器的当前值。

参考信息：Util 帮助程序对象

实用函数

名称	参数	返回值类型	注释
cleanString	字符串值	字符串	用于对必须在 XML 文档中转义的任何输入字符进行转义。 示例： <pre><Value>@{util.cleanString(p root/row/ a_value')}}</ Value></pre>

名称	参数	返回值类型	注释
getExtForContentType	字符串 MIME 类型	字符串	使集成能够根据指定的 MIME 类型获取文件扩展名，例如 .txt 或 .xml。
getFilenameFromContentDisposition	字符串 content-disposition	字符串	从 content-disposition 字符串中提取文件名。某些程序集组件（例如文档迭代器策略）会使用文件名信息填充内容处置消息标头。 示例：此代码返回值 testfile.pdf： <pre>util.getFilenameFromContentDisposition(filename=testfile.pdf')</pre>
readFileToVar	String var_name, String 文件名		允许将非文本文件读取到变量中。这是所提供的加载非文本文档的唯一方法。请注意，“write”步骤的 static-file 属性只能将文本文档加载到消息或变量中。文件是相对于 WSAR-INF 目录指定的。 示例：以下代码会将图片文件 icon-32.png 从 WSAR-INF 目录读取到上下文变量 v1 中： <pre>util.readFileToVar('v1', 'icon-32.png')</pre>
readFileToVar	String var_name,String filename,String MIMETYPE		允许将非文本文件读取到您指定其 MIME 类型的变量中。这是所提供的加载非文本文档的唯一方法。请注意，“write”步骤的 static-file 属性只能将文本文档加载到消息或变量中。文件是相对于 WSAR-INF 目录指定的。 示例：以下代码将图片文件 icon-32.png 从 WSR-INF 目录读取到上下文变量 v1 中，多

名称	参数	返回值类型	注释
			用途互联网邮件扩展类型为“img/png”： util.readFileToVar('v1', 'ico.png')

参考信息：Workday Studio MVEL 表达式示例

可变	表达式示例	描述
context	context.getProperty('myProperty')	检索属性的值 myProperty 从 MediationContext。
	context.getProperty('someProperty').equals('someValue')	检索属性 someProperty 从 MediationContext 如果属性值设置为 someValue。
da	da.toVar('eib_transform_output.txt', 'var1')	从集成事件中检索文件名为 "eib_transform_output.txt" 的

可变	表达式示例	描述
		文档，并添加名为""的变量。 var1。
	<code>da.getDocsMatching('label')</code>	检索具有给定标签的文档。
ec	<code>ec.getAttribute('attributeName')</code>	检索属性名称。
	<code>ec.integrationMapLookup('map', 'key')</code>	检索集成映射值。
env	<code>env['cc.hostname']</code>	检索 cc.hostname 环境属性设置为 -Dcc.hostname 请在""行中选择""。
	<code>env.containsKey('key')</code>	如果环境属性包含给定的键，则显示为 true。
ftp	<code>ftp.list('URL')</code>	检索 URL 对应的文件列表。
intsys	<code>intsys.getAttribute('username')</code>	检索名为""的集成属性。 username。
	<code>intsys.isServiceEnabled('name')</code>	如果存在具有指定名称的集成服务，则此选项显示为“真”。
lp	<code>lp.getSimpleData('name')</code>	检索具有给定名称的启动参数的值。
	<code>lp.exists('label') == true</code>	如果存在具有给定标签的参数，则显示为真。
message	<code>message.rootPartAsText</code>	从""中检索文本形式的消息根部分 MediationMessage 接口。
	<code>message.variables</code>	检索MediationContext 变量。
mtable	<code>mtable['my_property'].set(0, 'Data_Column', 'my_test_data')</code>	在报告中设置数据 mtable。
	<code>mtable['my_property'].get(0, 'Data_Column')</code>	从检索数据 mtable。
parts	<code>parts[0].getHeader('someHeader')</code>	检索标头 someHeader 从根部件开始
	<code>parts[0].xpathB('/a/b')</code>	实现 xpathB 方法来评估 XPath 布尔表达式。
props	<code>props.myProperty</code>	检索属性的值 myProperty 从 MediationContext。
	<code>props['bean.property']</code>	检索属性的值 bean.property 从 MediationContext。
spring	<code>spring.getBean('fred').name</code>	检索 name 名为""的 bean 的属性 Fred 在
	<code>spring.parent ApplicationContext</code>	访问 Spingle 应用程序上下文。 Workday Proprietary and Confidential

参考信息：Workday Studio MVEL 自动转换语法示例

函数	表达式示例	描述
将字符串自动转换为变量。	<pre>vars['fred'] = '<hello>world</ hello>'</pre>	创建一个名为“”的新变量 fred 与内容 <hello>world</hello>。 使用此方法仅创建短消息。使用表达式创建大型字符串可能会导致性能降低。 在大多数情况下，您应该使用 write 步骤 为变量创建内容。
为变量提供内容和多用途互联网邮件扩展 (MIME) 类型信息。	<pre>vars['fred'] = ['Hello World', 'text/plain']</pre>	创建一个名为“”的新变量 fred 与内容 Hello World 和 MIME 类型 text/plain。 使用此方法仅创建短消息。使用表达式创建大型字符串可能会导致性能降低。 在大多数情况下，您应该使用 write 步骤 为变量创建内容并配置 MIME 类型。
从附件复制到变量。	<pre>vars['fred'] = parts[1]</pre>	创建一个新变量 fred 从当前消息的第一个附件 在大多数情况下，您应该使用 copy 以下步骤可从附件获取输入。
从消息复制到变量。	<pre>vars['fred'] = message</pre>	创建一个新变量 fred 整封邮件中的“”。 在大多数情况下，您应该使用 copy 步骤：从当前消息获取输入
向值为表达式的变量提供内容和 MIME 类型信息。	<pre>vars['fred'] = [23 + 5, 'text/plain']</pre>	创建一个名为“”的新变量 fred 有内容 28 和 MIME 类型 text/plain。
根据消息根部件中的 XPath 片段创建一个变量。	<pre>vars['fred'] = parts[0].xpathF('/ dink/donk')</pre>	创建一个名为“”的新变量 fred 包含当前消息根目录中的 XPath 片段。 使用此方法仅创建短消息。使用 XPath 表达式创建大型字符串可能会导致性能不佳。
将变量设为 null。	<pre>vars['fred'] = null</pre>	移除名为“”的变量 fred。
将内容指定给根部分。	<pre>parts[0] = '<hello>world</ hello>'</pre>	指定内容 <hello>world</hello> 添加到根部分。 使用此方法仅创建短消息。使用表达式创建大型字符串可能会导致性能降低。

函数	表达式示例	描述
		在大多数情况下，您应该使用 write 步骤：为附件指定内容并配置多用途互联网邮件扩展 (MIME) 类型
为第一个附件指定内容。	<pre>parts[1] = ['Hello World', 'text/plain']</pre>	指定内容 Hello World 和 MIME 类型 text/plain 添加到第一个附件。 使用此方法仅创建短消息。使用表达式创建大型字符串可能会导致性能降低。 在大多数情况下，您应该使用 write 步骤：为附件指定内容并配置多用途互联网邮件扩展 (MIME) 类型
为根附件指定内容并指定多用途互联网邮件扩展 (MIME) 类型。	<pre>parts[0] = [23 + 5, 'text/plain']</pre>	为根部件指定内容 28 和 MIME 类型 text/plain。
将变量中的内容指定到消息部分。	<pre>parts[0] = vars['fred']</pre>	为变量的内容指定值 fred 添加到消息部件 在大多数情况下，您应该使用 copy 步骤，从变量获取输入并将其指定给消息部件。
移除根部分。	<pre>parts[0] = null</pre>	移除根部分。
复制附件。	<pre>parts[2] = parts[1]</pre>	将第一个附件复制到第二个附件。 在大多数情况下，您应该使用 copy 步骤，从一个附件获取输入并将其指定给另一个附件。
使用默认的多用途互联网邮件扩展 (MIME) 类型创建新消息。	<pre>context.message = '<hello>world</hello>'</pre>	创建新消息 <hello>world</hello> 其默认 MIME 类型为 text/xml。 在大多数情况下，您应该使用 write 步骤“”以创建新消息。
创建多用途互联网邮件扩展 (MIME) 类型为新消息 text/plain。	<pre>context.message = ['Hello World', 'text/plain']</pre>	创建新消息 Hello World MIME 类型为 text/plain。 在大多数情况下，您应该使用 write 步骤“”以创建新消息。
根据可变内容创建新消息。	<pre>context.message = vars['fred']</pre>	创建与变量具有相同内容和 MIME 类型的新消息 fred。 在大多数情况下，您应该使用 copy 执行步骤，从变量创建新消息。
从当前消息的根部分创建新消息。	<pre>context.message = parts[0]</pre>	从当前消息的根部分创建新消息。

函数	表达式示例	描述
		在大多数情况下，您应该使用 copy 步骤：根据消息部件创建新消息。

参考信息：MVEL 方法

方法	描述	示例代码	示例输出
<code>getClass().getName()</code>	检索属性或变量的类型。	<code>props['p.myProperty']</code> <code>props['p.myProperty'].getClass().getName()</code>	<code>java.lang.Boolean</code>
<code>getEndpoint()</code>	使用 SOAP 请求时，检索 RaaS 报告的端点。	<code>@{intsys.reportServiceAlias'}}</code>	<code>customreport2/Report</code> <code><tenant_name>/</code> <code><username>/</code> <code>report_name_in_XML_format</code>
<code>getExtrapath()</code>	检索 RaaS 报告的额外路径。	<code>intsys.reportService</code> <code>workday-in</code> <code>Report Service</code> <code>Alias>')</code>	<code>customreport2/your</code> <code><tenant_name>/</code> <code><username>/</code> <code><Custom Report</code> <code>Name></code>
<code>indexOf()</code>	检索特定字符在输入字符串中的位置。	<code>props['input.document']</code> <code>=</code> <code>'FoundationDataCollection/4d196113'</code> <code>props['index.of.separator']</code> <code>=</code> <code>props['input.document.reference'].indexOf("/")</code>	<code>24</code>
<code>leftPad()</code>	在输入字符串的开头添加字符。	<code>org.apache.commons</code> <code>10, '0')</code>	<code>0000001234</code> <code>leftPad('1234', 10, '0')</code>
<code>listToCommaDelimitedString()</code>	将数组值列表转换为字符串，并用逗号分隔这些值。	<code>props[p.workers]</code> <code>= [1234, 7890]</code> <code>props['p.workers']</code> <code>=</code> <code>lp.getWIDs('Pay</code> <code>Group</code> <code>Members')</code> <code>props['p.workers.ids']</code> <code>=</code> <code>props['p.workers'] !</code> <code>= empty ?</code> <code>util.listToCommaDelimitedString(props['p.workers'])</code>	<code>1234, 7890</code>
<code>replaceAll()</code>	替换输入字符串中所有指定字符的实例。	<code>props['p.employee.id']</code> <code>=</code> <code>'12345pdf_a.pdf'</code>	<code>12345</code>

方法	描述	示例代码	示例输出
		<pre>props['p.current.id'] = props['p.employee.id'].replaceAll('[^0-9]', '')</pre>	
substring()	检索部分输入字符串。	<pre>props['p.test.string'] = 'test' props['p.test.substring'] = props['p.test.string'].substring(0,2)</pre>	
toUpperCase()	将字符串转换为大写。	<pre>props['p.test.string'] = 'workday Studio' props['p.test.string'].toUpperCase()</pre>	WORKDAY STUDIO
URLEncoder ()	检索仅包含有效 URL 字符的字符串。	<pre>props['p.report.parameters'] = 'Full_Name=' # 'Logan McNeil' props['REST.URL'] = java.net.URLEncoder.encode(props['p.report.parameters'], 'UTF-8')</pre>	Full_Name=Logan %20McNeil

Workday Studio 集成服务

创建属性映射服务

开始之前

创建一个组件项目。

关于此任务

通过属性映射服务，您可以为集成定义属性和映射。属性是用于定义集成的可配置值。您可以创建以下类型的属性：

- 简单
- 引用
- 枚举

映射会将内部 Workday 值与集成中定义的外部属性值进行核对。您可以创建以下类型的映射：

- 简单
- 类报告字段
- 枚举

过程

1. 在 Assembly Editor 中打开程序集。
2. 将一个“workday-in”传输从“Palette”拖动到 Assembly Editor 中。
3. 在该“workday-in”传输的“Properties”视图的“Services”选项卡上，单击“Create attribute-map-service”。

4. 在“Service Name”窗口中，输入属性映射服务的名称。
5. 在“Integration Attributes”选项卡中，请参考以下信息：

选项	描述
名称	属性的名称。在所有集成系统中必须是唯一的。
类型	属性类型定义。
Add Simple Type Attribute	添加使用字符串字面值的属性类型。
Add Reference Type Attribute	添加一个属性类型，该类型使用对 Workday 类报告字段 (CRF) 和 Web 服务的引用。
Add Enumeration Type Attribute	选择“Edit Type”以配置枚举。您可以： - 输入相应类型名称，以便使用租户中某个现有枚举定义的枚举引用。 - 在“Enumeration Definition”选项卡中输入一个唯一的类型名称，并指定其允许的值，以便新建枚举。 注：当您在“Enumeration Definition”选项卡中为现有定义输入类型名称时，Workday 会重设该定义。

6. 在“Integration Maps”选项卡上，选择“Add”以添加映射。
7. 配置“Internal Type”和“External Type”时，请参考以下信息：

选项	描述
Simple Type	您可以选择以下简单类型： - 布尔值 - 日期 - Datetime - 数字 - 文本
Class Report Field	选择“Browse for CRF”。
Enumeration	配置您的枚举引用或枚举定义。

结果

Studio 会在您集成的 Workday-in 传输上保存属性映射服务。您可以使用 MVEL 从服务中访问数据 intsys 表达式。

下一步做什么

部署集成

relatedlinks

概念

[Concept: Enumerations](#)

任务

[将集成部署到 Workday on page 254](#)

[设置福利集成中的属性和映射](#)

创建传送服务

开始之前

创建一个程序集项目。

关于此任务

通过传送服务，您可以将文档从 Workday 租户传送到目标端点。

过程

1. 在 Assembly Editor 中打开程序集。
2. 在“Palette”中，展开“In Transports”选项卡。
3. 将一个“workday-in”传输从“Palette”拖动到 Assembly Editor 中。
4. 在 Assembly Editor 中，单击该“workday-in”传输。
5. 在该“workday-in”传输的“Properties”视图的“Services”选项卡上，单击“Create delivery-service”。
6. 在“Service Name”窗口中，输入传送服务的名称。
7. 单击“保存”。

结果

Workday Studio 会将该传送服务保存在集成的“workday-in”传输中。

下一步做什么

在 Workday 中部署集成，并为您的集成系统配置传送服务。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[设置集成交付](#)

创建侦听器服务

开始之前

创建一个程序集项目。

关于此任务

通过侦听器服务，您的集成能够通过您的 Workday 租户接收来自第三方客户端的启动消息。

过程

1. 在 Assembly Editor 中打开程序集。
2. 在“Palette”中，展开“In Transports”选项卡。
3. 将一个“workday-in”传输从“Palette”拖动到 Assembly Editor 中。
4. 在 Assembly Editor 中，单击该“workday-in”传输。
5. 在该“workday-in”传输的“Properties”视图的“Services”选项卡上，单击“Create listener-service”。
6. 在“Service Name”窗口中，输入侦听器服务的名称。
7. 单击“保存”。

结果

Workday Studio 会将该侦听器服务保存在集成的“workday-in”传输中。

下一步做什么

在 Workday 中部署集成，并为您的集成系统配置侦听器服务。

注：对于使用侦听器服务的集成，仅当其没有任何启动参数时，您才能进行部署。

relatedlinks

概念

[概念：侦听器服务 API](#)

任务

[将集成部署到 Workday on page 254](#)

创建检索服务

开始之前

创建一个程序集项目。

关于此任务

通过检索服务，您的集成可以从远程客户端检索文档，然后将文档存储在您的 Workday 租户中。

过程

1. 在 Assembly Editor 中打开程序集。
2. 在“Palette”中，展开“In Transports”选项卡。
3. 将一个“workday-in”传输从“Palette”拖动到 Assembly Editor 中。
4. 在 Assembly Editor 中，单击该“workday-in”传输。
5. 在该“workday-in”传输的“Properties”视图的“Services”选项卡上，单击“Create retrieval-service”。
6. 在“Service Name”窗口中，输入检索服务的名称。
7. 单击“保存”。

结果

Workday Studio 会将检索服务保存在集成的“workday-in”传输中。

下一步做什么

在 Workday 中部署集成，并为您的集成系统配置检索服务。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[设置集成检索](#)

创建序列生成器服务

开始之前

创建一个程序集项目。

关于此任务

通过序列生成器服务，您可以在每次运行集成时为文档生成唯一的有序文件名。在 Workday Studio 中，您可在“workday-in”传输中配置序列生成器服务。在启动集成之前，您还必须在 Workday 租户中配置序列生成器服务。

过程

- 1. 在 Assembly Editor 中打开程序集。
- 2. 在 Assembly Editor 中，单击“workday-in”传输。
- 3. 在“workday-in”传输的“Properties”视图的“Services”选项卡上，单击“Create sequence-generator-service”。
- 4. 在“Service Name”窗口中，输入序列生成器服务的名称。
- 5. 在序列生成器服务的“Properties”视图中，单击“Add”以添加排序器。
- 6. 在“Sequencer”字段中，输入排序器的名称。
- 7. 在“Project Explorer”视图中，右键单击程序集项目，选择“Deploy to Workday”。
- 8. 在“Cloud Explorer”视图中，导航到已部署的程序集项目。
- 9. 右键单击已部署的程序集项目的集成节点。
- 10. 选择“Workday: Integration System”。
- 11. 在“Workday”选项卡上，请参考以下信息：

选项	描述
Username	Workday 租户的用户名凭据。
Password	租户的密码凭据。

- 12. 在您的 Workday 租户中，在“自定义集成 服务”板块的“集成 服务”列中，找到您的序列生成器服务的名称，然后依次选择 “相关操作”和“预览”。
- 13. 选择 “集成序列生成器服务” > > “配置集成序列生成器”。
- 14. 在 “序列编号的增量依据” 字段中，输入值“1”。
- 15. 在 “序列编号格式” 字段中，输入值 file[Seq].csv。
- 16. 单击“OK”。
- 17. 单击“Done”。

结果

Workday Studio 会将序列生成器服务保存在集成的“workday-in”传输中。Workday 会将序列生成器设置保存在您的集成系统中。

下一步做什么

启动您的集成。

relatedlinks

任务

[通过 Workday Studio 启动集成](#) on page 260

[设置集成序列生成器](#)

创建事务日志服务

开始之前

创建一个程序集项目。

关于此任务

使用事务日志服务，您可在 Workday 租户中显示您的集成已订阅的事务事件日志。

过程

- 1. 在 Assembly Editor 中打开程序集。
- 2. 在“Palette”中，展开“In Transports”选项卡。
- 3. 将一个“workday-in”传输从“Palette”拖动到 Assembly Editor 中。
- 4. 在 Assembly Editor 中，单击该“workday-in”传输。
- 5. 在该“workday-in”传输的“Properties”视图的“Services”选项卡上，单击“Create transaction-log-service”。
- 6. 在“Service Name”窗口中，输入事务日志服务的名称。
- 7. 单击“保存”。

结果

Workday Studio 会将事务日志服务保存在集成的“workday-in”传输中。

下一步做什么

在 Workday 中部署集成，并为您的集成系统配置事务日志服务。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[步骤：设置目录服务集成](#)

创建报告服务

开始之前

创建一个程序集项目。

关于此任务

报告服务会为您在集成中引用的所有自定义报告指定别名。

过程

- 1. 在 Assembly Editor 中打开程序集。
- 2. 在“Palette”中，展开“In Transports”选项卡。
- 3. 将一个“workday-in”传输从“Palette”拖动到 Assembly Editor 中。
- 4. 在 Assembly Editor 中，单击该“workday-in”传输。
- 5. 在该“workday-in”传输的“Properties”视图的“Services”选项卡上，单击“Create report-service”。
- 6. 在“Service Name”窗口中，输入报告服务的名称。
- 7. 在报告服务的“Properties”视图中，单击“Add New Entry”以添加新的别名。完成此任务时，请参考以下信息：

选项	描述
别名	自定义报告的别名。
描述	自定义报告的文本描述。
Report Reference	自定义报告在您的 Workday 租户中显示的名称。

8. 单击“Select a RaaS Report”。完成此任务时，请参考以下信息：

选项	描述
环境	包含您的自定义报告的 Workday 环境。
ReportName (报告名称)	自定义报告的名称。

- 9. 单击“下一步”。
- 10.选择您的自定义报告。
- 11.单击“Finish”。
- 12.单击“保存”。

结果

Workday Studio 会将报告服务保存在集成的“workday-in”传输中。

下一步做什么

在 Workday 中部署集成，并为您的集成系统配置报告服务。

relatedlinks

概念

[概念：自定义报告](#)

任务

[将集成部署到 Workday on page 254](#)

创建自定义对象服务

开始之前

创建一个程序集项目。

关于此任务

自定义对象服务会为您在集成中引用的任何自定义对象指定别名。

过程

- 1. 在 Assembly Editor 中打开程序集。
- 2. 在“Palette”中，展开“In Transports”选项卡。
- 3. 将一个“workday-in”传输从“Palette”拖动到 Assembly Editor 中。
- 4. 在 Assembly Editor 中，单击该“workday-in”传输。
- 5. 在该“workday-in”传输的“Properties”视图的“Services”选项卡上，单击“Create custom-object-service”。
- 6. 在“Service Name”窗口中，输入自定义对象服务的名称。
- 7. 在自定义对象服务的“Properties”视图中，单击“Add New Entry”以添加新的别名。完成此任务时，请参考以下信息：

选项	描述
别名	自定义对象的别名。
描述	自定义对象的文本描述。
Custom Object Reference	自定义对象在您的 Workday 租户中显示的名称。

- 8. 单击“Select a Custom Object”。

9. 在“Select Custom Object”窗口中，指定您的 Workday 租户环境，然后浏览到您的自定义对象。
10. 单击“OK”。
11. 单击“保存”。

结果

Workday Studio 会将自定义对象服务保存在集成的“workday-in”传输中。

下一步做什么

在 Workday 中部署集成，并为您的集成系统配置自定义对象服务。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[设置集成自定义对象](#)

创建数据初始化服务

开始之前

创建一个程序集项目。

关于此任务

通过数据初始化服务，您可以更快速、更高效地访问 Workday Studio 集成中的 Workday 数据。

过程

1. 在 Assembly Editor 中打开程序集。
2. 在“Palette”中，展开“In Transports”选项卡。
3. 将一个“workday-in”传输从“Palette”拖动到 Assembly Editor 中。
4. 在 Assembly Editor 中，单击该“workday-in”传输。
5. 在该“workday-in”传输的“Properties”视图的“Services”选项卡上，单击“Create data-initialization-service”。
6. 在“Service Name”窗口中，输入数据初始化服务的名称。
7. 单击“保存”。

结果

Workday Studio 会将该数据初始化服务保存在集成的“workday-in”传输中。

下一步做什么

在 Workday 中部署集成，并为您的集成系统配置数据初始化服务。

创建服务引用

开始之前

创建一个程序集项目。

关于此任务

通过服务引用，您可以引用在同一集合内的其他“workday-in”传输中配置的任何集成服务。

过程

1. 在 Assembly Editor 中打开程序集。
2. 在“Palette”中，展开“In Transports”选项卡。
3. 将一个“workday-in”传输从“Palette”拖动到 Assembly Editor 中。
4. 在 Assembly Editor 中，单击该“workday-in”传输。
5. 在该“workday-in”传输的“Properties”视图的“Services”选项卡上，单击“Create service-reference”。
6. 在“Service Reference”窗口中，选择一个服务引用。
7. 单击“OK”。
8. 单击“保存”。

结果

Workday Studio 会将该服务引用保存在集成的“workday-in”传输中。

下一步做什么

部署您的集成。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

在 **Workday-Out-Rest** 传输上添加对自定义报告的引用

开始之前

创建一个包含“workday-out-rest”传输的程序集项目。

关于此任务

通过“workday-out-rest”传输，您可将 REST 请求消息发送到 HTTP URL。您可以添加对自定义报告的引用，将其作为“workday-out-rest”传输上的额外路径值。

过程

1. 在 Assembly Editor 中打开程序集。
2. 在 Assembly Editor 中，单击“workday-out-rest”传输。
3. 在“workday-out-rest”传输的“Properties”视图的“Common”选项卡上，单击“Select a RaaS Report”。
4. 在“Add Custom RaaS Report”窗口中，请参考以下信息：

选项	描述
环境	包含您的自定义报告的 Workday 环境。
ReportName (报告名称)	自定义报告的名称。

5. 在“Select Reports”窗口中，选择您的自定义报告。
6. 在“Report Service Configuration”窗口中，请参考以下信息：

选项	描述
Report Alias	自定义报告的别名。
描述	自定义报告的文本描述。

7. 单击“Finish”。

结果

Workday Studio 会在集成的“workday-out-rest”传输中添加对自定义报告的引用。

下一步做什么

启动您的集成。

relatedlinks

任务

[通过 Workday Studio 启动集成](#) on page 260

在 **Workday-Out-Rest** 传输上添加对自定义对象的引用

开始之前

创建一个包含“workday-out-rest”传输的程序集项目。

关于此任务

通过“workday-out-rest”传输，您可将 REST 请求消息发送到 HTTP URL。您可以添加对自定义对象的引用，将其作为“workday-out-rest”传输上的额外路径值。

过程

1. 在 Assembly Editor 中打开程序集。
2. 在 Assembly Editor 中，单击“workday-out-rest”传输。
3. 在“workday-out-rest”传输的“Properties”视图的“Common”选项卡上，单击“Select a Custom Object”。
4. 在“Add Custom Object”窗口中，请参考以下信息：

选项	描述
环境	包含您的自定义对象的 Workday 环境。
HTTP Methods	选项包括： • POST • PUT • GET • DELETE

5. 在“Select Custom Object”窗口中，请参考以下信息：

选项	描述
Select an existing custom object alias	显示现有自定义对象别名的列表。
Create a new custom object alias	允许您新建自定义对象别名。

如果您选择了“PUT”“GET”或“DELETE”HTTP 方法，请跳至步骤 7。

6. 如果您选择了“POST”HTTP 方法，则 Workday Studio 会显示“POST Query Parameters”窗口。完成此任务时，请参考以下信息：

选项	描述
updateIfExists	更新现有自定义对象。
bulk	创建或更新自定义对象的集合。

7. 如果您选择了“PUT”“GET”或“DELETE”HTTP 方法，请从“Custom Object Id Reference Selection”窗口中选择一个自定义对象编号引用。
8. 单击“Finish”。

结果

Workday Studio 会在集成的“workday-out-rest”传输中添加对自定义对象的引用。

下一步做什么

启动您的集成。

relatedlinks

任务

[通过 Workday Studio 启动集成](#) on page 260

CLAR 文件

生成清单文件

关于此任务

您可以利用清单文件，将程序集集合导出为 CLAR 文件。您必须将清单文件保存到程序集集合的项目文件夹中。清单文件包含程序集集合的元数据。

过程

1. 在 Workday Studio 菜单栏上，选择“Workday” > “Solution” > “Generate Workday Manifest”。
2. 双击清单文件的目标项目文件夹。

结果

Workday Studio 会将清单文件保存到该项目文件夹中。

下一步做什么

将程序集集合导出为 CLAR 文件。

导入 CLAR 文件

开始之前

将 CLAR 文件保存到文件系统。

关于此任务

可以通过导入 CLAR 文件将程序集添加到工作区。

过程

1. 在 Studio 菜单栏上，选择“文件” > “导入...”。
2. 选择 CLAR 文件 导入向导。
3. 单击“Browse”，然后选择要导入的 CLAR 文件。
4. 单击“Next”以显示 CLAR 文件中的集合和项目。
5. 选择要从 CLAR 文件导入的集合和项目。

Workday Studio 会自动选择 CLAR 文件中的每一个集合和项目。您可使用复选框修改所选项。如果集合和项目存在名称冲突，Workday Studio 会为其显示修饰符。

6. 通过重命名存在名称冲突的项目来解决所有项目名称冲突。
 - a) 选择存在名称冲突的项目。
 - b) 单击“Edit...”。
 - c) 在“Project Name”字段中，为项目重命名。
7. 编辑所有存在名称冲突的集合。
 - a) 选择存在名称冲突的集合。
 - b) 单击“Edit...”。
 - c) 在“Collection Name”字段中，为集合重命名。
8. 单击“Finish”。

结果

Workday Studio 会将 CLAR 文件的内容导入到工作区。

导出 CLAR 文件

开始之前

- 创建一个程序集项目。
- 生成一个清单文件。

关于此任务

您可以将工作区中的程序集导出为 CLAR 文件。您可以使用 CLAR 文件与其他 Workday Studio 集成开发者共享程序集。例如：将它们上传到 Workday Community 的“贡献的解决方案”页面。

过程

1. 在 Studio 菜单栏上，选择“文件”>“导出...”。
2. 选择 CLAR 文件 导出向导。
3. 在“CLAR File Specification”窗口中，选择要导出为 CLAR 文件的集合。
4. 在“Workday Manifest Specification”窗口中，选择“Use existing manifest from collection”。
5. 在“Export Destination”窗口中，指定 CLAR 文件的目标位置。
6. 单击“Finish”。

结果

Workday Studio 会将这些程序集导出为 CLAR 文件。

relatedlinks

参考信息

[All Contributed Solutions](#)

概念：CLAR 文件

CLAR 文件

CLAR (云存档) 文件是一种打包文件，用于存储程序集集合、项目和元数据。利用 CLAR 文件，您可以通过工作区导入和导出程序集。您可以使用 CLAR 文件与其他 Workday Studio 集成开发者共享程序集。

清单文件

您可以利用清单文件，将程序集集合导出为 CLAR 文件。您必须将清单文件保存到程序集集合的项目文件夹中。清单文件包含程序集集合的元数据。

Contributed Solutions

“Contributed Solutions”页面是一个在线存储库，您可以在其中访问和共享 CLAR 文件。您可以在 Workday Community 中访问“Contributed Solutions”页面。

JAXB 项目

创建 JAXB 项目

开始之前

将 XML 架构文件保存到您的文件系统。

关于此任务

JAXB 是一个绑定框架，可让您在 XML 格式与 Java 格式之间转换数据。在 Workday Studio 中，您可以通过创建 JAXB 项目将 XML 架构文件转换为 Java 类。创建 JAXB 项目后，您可以通过将其添加为程序集项目中的依赖项，在集成中实施其 Java 类。

- 过程
1. 选择“File” > “New” > “Workday Schema to Java (JAXB) Project”。
 2. 在“项目配置详细信息”窗口中，请参考以下信息：

选项	描述
Project name	项目的名称。
Target runtime	项目的目标运行时服务器。自动填充为“Workday Runtime”。
配置	项目的分面配置。自动填充为“Workday Schema to Java (JAXB) Project”。

3. 在“Source Folders”窗口中，请参考以下信息：

选项	描述
Generation directory	JAXB 生成的 Java 类的目录文件夹。自动填充为“jaxb-gen”。
Mark generated resources as derived	将 JAXB 生成的 Java 类标记为派生资源。Workday Studio 会自动选中此复选框。

4. 在“JAXB Schema Selection and Generation Settings”窗口中，请参考以下信息：

选项	描述
Schema Files	XML 架构文件的位置。Workday Studio 会显示以下选项： • Workspace • 外部 • URL
JAXB External Binding Files (optional)	XML 架构的任何绑定文件的位置。Workday Studio 会显示以下选项： • Workspace • 外部

选项	描述
Generate default binding file for unmapped schemas	为您的 XML 架构创建通用绑定文件。
Location within Project	通用装订文件所在的目录文件夹。自动填充为 jaxb-resources/bindings.xjb。
选择如何管理架构文件和外部 JAXB 绑定文件	选择“Create links in the project to external Schema/Binding files”会创建链接（指向新 JAXB 项目中的架构和绑定文件）。Workday Studio 会自动选择此选项。 选择“Copy Schema/Binding files to project now”后，架构和绑定文件会被复制到新的 JAXB 项目中。
Compiler Options	Workday Studio 会显示以下选项： - Specify target package (overrides all other package definitions) - Apply less strict validation to schemas - Suppress package level annotations - Automatic name conflict resolution

5. 单击“Finish”。

结果

Workday Studio 会在“Project Explorer”视图中显示 JAXB 项目。

下一步做什么

将 JAXB 项目添加为程序集项目中的前序项。

将 JAXB 项目添加为程序集项目中的前序项

开始之前

- 创建一个程序集项目。
- 创建一个 JAXB 项目。

关于此任务

JAXB 是一个绑定框架，可让您在 XML 格式与 Java 格式之间转换数据。在 Workday Studio 中，通过将 JAXB 项目添加为程序集项目中的依赖项，您可以在集成中实现 JAXB Java 类。

过程

1. 在“项目浏览器”视图中，右键单击一个装配件项目。
2. 选择“构建路径” > “配置构建路径...”。
3. 在“Java Build Path”窗口中，选择“Projects”选项卡。
4. 单击“Add...”。
5. 选择要添加到程序集项目中的 JAXB 项目。
6. 单击“OK”。
7. 单击“Apply and Close”。

结果

Workday Studio 会将 JAXB 项目添加为程序集项目中的依赖项。现在，您可以在该程序集项目内存储的集成中实现 JAXB Java 类。

下一步做什么

部署您的集成。

概念：Workday Studio 中的 JAXB 支持

JAXB 是一个绑定框架，可让您在 XML 格式与 Java 格式之间转换数据。

JAXB 项目

在 Workday Studio 中，您可以通过创建 JAXB 项目将 XML 架构文件转换为 Java 类。创建 JAXB 项目后，您可以通过将其添加为程序集项目中的前序项，在集成中实现其 Java 类。

JAXB 中介步骤

在程序集项目中，您可以在以下中介步骤的“Properties”视图中调用 JAXB：

- java-to-xml
- xml-to-java

Workday Studio Ivy 插件

配置 Ivy 插件

关于此任务

Workday Studio 支持 Apache Ivy™ 插件，让您能够管理项目依赖关系。您可以将 Ivid 插件配置为 Studio 的首选项。

过程

1. 单击“窗口” > “首选项” > “Workday” > “Ivy”。
2. 在“Configure Ivy Engine”窗口中，请参考以下信息：

选项	描述
Ivy settings URL	ivysettings.xml 文件的 URL。
Ivy Properties	ivyproperties.xml 文件的 URL。
Load ivy.properties file from home directory if present	自动从您的主目录加载 ivy.properties 文件。

结果

Workday Studio 会将您的 Ivy 插件首选项应用于您的程序集项目。

在 Workday Studio 项目中添加 Workday Ivy 特性

开始之前

- 配置适用于 Workday Studio 的 Apache Ivy™ 插件。
- 创建一个程序集项目。

关于此任务

Workday Studio 支持 Apache Ivy 插件，这让您能配置具有 Workday Ivy 特性的项目，以便管理项目依赖关系。

过程

- 1. 在“项目浏览器”视图中，右键单击一个装配件项目。
- 2. 选择“Configure” > “Add Workday Ivy Nature”。
- 3. 展开程序集项目文件夹。
- 4. 打开 ivy.xml 文件。
- 5. 展开 信息 节点。完成此任务时，请参考以下信息：

选项	描述
Module	模块名称。
在此处添加您的内容	插入父级 <dependencies> 元素（包含一个或多个子级）<dependency> 元素。

结果

Studio 会为项目添加 Workday Ivo 性质。

解析具有 Ivy 定义的项目的依赖关系

开始之前

- 配置适用于 Workday Studio 的 Apache Ivy™ 插件。
- 配置具有 Workday Ivy 特性的程序集项目。

关于此任务

Workday Studio 支持 Apache Ivy 插件，这让您能配置具有 Workday Ivy 特性的项目，以便管理项目依赖关系。如果您的项目显示构建错误，并且此错误与缺少依赖关系有关，您可以解析这些依赖关系问题。

过程

- 1. 在“Project Explorer”视图中，右键单击一个具有 Workday Ivy 特性的程序集项目。
- 2. 选择“Resolve”。

结果

“Console”视图会显示解析报告。

向具有 Ivy 特性的项目添加从属 JAR

开始之前

- 配置适用于 Workday Studio 的 Apache Ivy™ 插件。
- 配置具有 Workday Ivy 特性的程序集项目。

关于此任务

Workday Studio 支持 阿普奇 (Workday) 插件，该插件使您能够通过配置具有 Workday Ivid 性质的项目来管理项目依赖关系。如果 Studio 项目具有 Ivo 性质，且其 ivy.xml 文件中，则可以配置项目以包括任何从属 JAR 文件。

过程

1. 在“项目浏览器”视图中，右键单击具有“Workday I典型”性质的程序集项目。
2. 选择“Properties”。
3. 在“Java Build Path”窗口中，单击“Libraries”。
4. 单击“Ivy-Workday IDE”。
5. 单击“Add JARs”。
6. 在“JAR Selection”窗口中，选择要添加到项目的 JAR 文件，然后单击“OK”。
7. 在“Java Build Path”窗口中，单击“Apply and Close”。

结果

Workday Studio 会将从属 JAR 文件添加到您的项目中。

下一步做什么

部署项目。

从 Workday Studio 项目中移除 Workday Ivy 特性

开始之前

- 配置适用于 Workday Studio 的 Apache Ivy™ 插件。
- 配置具有 Workday Ivy 特性的程序集项目。

关于此任务

Workday Studio 支持 Apache Ivy 插件，这让您能配置具有 Workday Ivy 特性的项目，以便管理项目依赖关系。您可以按需添加和移除 Workday Ivy 特性。

过程

1. 在“项目浏览器”视图中，选择一个程序集项目文件夹。
2. 选择“Workday” > “Ivy” > “移除“Workday Ivy”性质”。

结果

Studio 会移除项目的 Workday Ive 性质。

下一步做什么

部署项目。

集成部署

添加连接

关于此任务

在 Workday Studio 中，您可以创建和管理与 Workday 的连接。在部署和启动程序集时，您可以使用这些连接。

过程

- 1. 选择“Window” > “Preferences” > “Workday” > “Connections”。
- 2. 单击“Add...”。
- 3. 在“Add Connection”窗口中，请参考以下信息：

选项	描述
名称	连接的名称。
URL	连接的 URL。
组	将连接添加到组。

- 4. 在“Credentials”版块中，指定租户并选择一种身份验证方法：“Basic Auth”或“OAuth 2.0”。
注: 要使用 OAuth 2.0 授权，您必须先注册 API 客户端。该客户端用户如已注册 Workday 多重身份验证，则在从 Workday Studio 连接到 Workday 时，必须使用第二种身份验证方法。

选项	描述
Basic Auth	指定连接的用户名和密码。
OAuth 2.0	为连接指定以下详细信息： • 授予类型 • 客户端 ID • 授权端点 • 令牌端点 • 访问令牌

- 5. 单击“Test Connection”。
Workday Studio 会确认能否访问该连接。

结果

该连接将显示在“Connections”首选项页面上。

下一步做什么

构建程序集。

relatedlinks

任务

[在 Workday Studio 中启用多重身份验证 on page 257](#)

添加连接组

关于此任务

在 Workday Studio 中，您可以创建和管理与 Workday 的连接。您可以将这些连接分组。部署和启动程序集时，您可以使用这些连接。

过程

- 1. 选择“Windows” > “首选项” > “Workday” > “连接”。
- 2. 单击“添加组...”。

3. 在“Add Group”窗口中，请参考以下信息：

选项	描述
名称	连接组的名称。
Filtered	从“Cloud Explorer”视图和“Deploy to Workday”页面中筛选出连接组。

结果

该连接组将显示在“Connections”首选项页面上。

下一步做什么

构建程序集。

编辑连接

开始之前

添加与 Workday 的连接。

关于此任务

在 Workday Studio 中，您可以创建和管理与 Workday 的连接。部署和启动程序集时，您可以使用这些连接。

过程

1. 选择“Windows” > “首选项” > “Workday” > “连接”。
2. 选择联系人。
3. 单击“Edit Environment URL”。完成此任务时，请参考以下信息：

选项	描述
Host	连接的主机的编号。
URL	连接的 URL。

结果

该连接将显示在“Connections”首选项页面上。

下一步做什么

构建程序集。

编辑连接组

开始之前

创建一个 Workday 连接组。

关于此任务

在 Workday Studio 中，您可以创建和管理与 Workday 的连接。您可以将这些连接分组。部署和启动程序集时，您可以使用这些连接。

过程

- 1. 选择“Windows” > “首选项” > “Workday” > “连接”。
- 2. 选择关联组。
- 3. 完成此任务时，请参考以下信息：

选项	描述
名称	连接组的名称。
Filtered	从“Cloud Explorer”视图和“Deploy to Workday”页面中筛选出连接组。

结果

该连接组将显示在“Connections”首选项页面上。

下一步做什么

构建程序集。

移除连接

开始之前

添加与 Workday 的连接。

关于此任务

在 Workday Studio 中，您可以创建和管理与 Workday 的连接。部署和启动程序集时，您可以使用这些连接。

过程

- 1. 选择“Windows” > “首选项” > “Workday” > “连接”。
- 2. 选择联系人。
- 3. 单击“Remove...”。

结果

Workday Studio 会从“Connections”首选项页面中移除该连接。

下一步做什么

构建程序集。

导入连接

开始之前

访问包含 Workday 连接详细信息的 XML 文件。

关于此任务

在 Workday Studio 中，您可以创建和管理与 Workday 的连接。部署和启动程序集时，您可以使用这些连接。

过程

1. 选择“Windows” > “首选项” > “Workday” > “连接”。
2. 单击“导入...”。
3. 单击“Browse...”。
4. 选择一个 Workday 连接 XML 文件。

结果

Workday Studio 会在“Connections”首选项页面上显示该连接。

下一步做什么

构建程序集。

导出连接

开始之前

添加与 Workday 的连接。

关于此任务

在 Workday Studio 中，您可以创建和管理与 Workday 的连接。部署和启动程序集时，您可以使用这些连接。

过程

1. 选择“Windows” > “首选项” > “Workday” > “连接”。
2. 单击“导出...”。
3. 选择一个连接。
4. 单击“Browse...”。
5. 为 Workday 连接 XML 文件选择一个目标位置。

结果

Workday Studio 会将连接导出到此目标位置。

下一步做什么

构建程序集。

连接 Workday

开始之前

添加与 Workday 的连接。

关于此任务

在 Workday Studio 中，您可以创建和管理与 Workday 的连接。部署和启动程序集时，您可以使用这些连接。

过程

1. 从 Studio“首选项”菜单中，选择“Workday > 连接”。
2. 选择联系人。完成此任务时，请参考以下信息：

选项	描述
租户	指定租户名称。
Username	输入租户的用户名凭据。
Password	输入租户的密码凭据。

3. 单击“Test Connection”。
Workday Studio 会确认能否访问您的租户。
4. 单击“Apply and Close”。

结果

Workday Studio 连接到您的 Workday 租户。

下一步做什么

将程序集部署到 Workday。

将集成部署到 Workday

开始之前

- 创建一个程序集项目。
- 安全性：“Integration”功能区域中的“Integration Build”域和“Integration Configure”域。

关于此任务

在 Workday Studio 中，您可以将集成部署到 Workday。在部署某个集成后，您可以在 Workday 租户中启动该集成。

过程

1. 在“Project Explorer”视图中，右键单击一个程序集项目。
2. 选择“Deploy to Workday”。
3. 在“Deploy to Workday”窗口中，请参考以下信息：

选项	描述
Workday Environments	选择用于部署集成的目标 Workday 环境。
Collection being deployed	选择要部署到 Workday 的集合。

4. 单击“Finish”。

结果

Workday Studio 会将您的集成部署到 Workday。

下一步做什么
启动您的集成。

查看已部署到 Workday 的集成

开始之前

- 将集成部署到 Workday。
- 安全性： “Integration”功能区域中的“Integration Build”域和“Integration Configure”域。

关于此任务

在 Workday Studio 中，您可以将集成部署到 Workday。您可以在登录 Workday 租户后查看已部署的集成。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Workday: Integration System”。
5. 在“Workday”选项卡上，登录您的 Workday 租户。完成此任务时，请参考以下信息：

选项	描述
Username	输入 Workday 租户的用户名凭据。
Password	输入租户的密码凭据。

结果

您已部署的集成将显示在 Workday Studio 的“Workday”选项卡上。

下一步做什么
启动您的集成。

配置 Studio 以使用 HTTPS 代理服务器

关于此任务

如果您需要通过 443 以外的端口将 Workday Studio 连接到训练或测试环境，并且您组织的防火墙默认不允许这样做，则也许可以使用网络中的代理服务器进行连接。

过程

1. 选择 “Windows > 首选项” > “常规” > > “网络连接”。
2. 从 “有效的提供商” 下拉菜单中，选择 “手动”。

注: 如果您选择 “手动”，则代理条目 表中的所有 3 个 架构（ HTTP、 HTTPS 和 SOCCS ）的复选框均处于选中状态。您无法逐一清除这些选择。
3. 在“HTTPS 架构”旁边，输入代理服务器的 主机 IP 地址和 端口 号。
4. 单击 “应用并关闭”。

概念：Workday Studio 连接

在 Workday Studio 中，您可以创建和管理与 Workday 的连接。在部署和启动程序集时，您可以使用这些连接。

注：所有连接名称必须是唯一的。Workday Studio 提供“Sandbox”和“Production”默认连接。您不能重命名这两个连接。出于安全原因，在您退出 Workday Studio 会话后，Workday Studio 不会将用户名和密码详细信息保存到您的工作区。

概念：Cloud Explorer

“Cloud Explorer”视图会显示您的云服务器连接。您可以展开每个云服务器节点来查看其集合。您可以展开每个集合节点，查看可启动的集成。

注：“Cloud Explorer”视图最多显示 999 个集合。

“Cloud Explorer”视图使用与本地操作系统相同的日期格式设置。

参考信息：连接错误

连接错误	注释
Your user login credentials failed to authenticate.	请检查您的凭据，然后重试。请注意，租户名称区分大小写。
Your tenant is not available.	使用 Web 浏览器检查您的租户是否在线。按以下格式输入租户 URL： <pre>https://host-name/tenant-name/ccx/cc-cloud-repo/collections</pre> 示例： <pre>https://wd5.myworkday.com/gms/ccx/cc-cloud-repo/collections</pre> 您的凭据与您尝试在 Studio 中使用的凭据相同，但要 will 将用户名称和租户名称与 @ 字符组合在一起。示例： <pre>lmcneil@gms</pre> 如果连接成功，浏览器将显示您的租户中部署的所有 Studio 集成集合。接下来，检查浏览器的连接首选项，看看浏览器是直接连接还是通过代理服务器进行连接。如果已将浏览器配置为使用代理服务器，则您也需要将 Workday Studio 配置为使用该代理。
Permission to deploy denied.	要将集成部署到 Workday，您必须属于“Integration Build”域中具有“修改”权限的安全组。请与您的 Workday 系统管理员联系。
Permission to launch denied.	要从 Workday Studio 启动集成，您必须属于以下某个域中具有“更新”权限的安全组： Integration Event

连接错误	注释
	- Integration Build - Integration Debug 请与您的 Workday 系统管理员联系。
Permission to deploy denied (integration contains a delivery or retrieval service).	要部署包含交付或检索服务的集成，您还需要一项额外的权限。您必须属于以下某个域中拥有“更新”权限的安全组： - Business Process Administration - Manage: Business Process Definitions - iLoad Web Services 请与您的 Workday 系统管理员联系。
Access to My Reports denied.	Workday 使用可配置的安全性来控制对“我的报告”集成文档的访问。您用于访问集成文档的 Workday 账户没有所需的权限。请与您的 Workday 系统管理员联系，申请所需的权限。
SOAP fault: A validation error has occurred.	您对 Web 服务操作的 SOAP 请求无效。确认您的请求是否包含有效的数据元素和值。 使用“Schema Explorer”中的“Add Latest Workday Web Services”向导，将有效的 WSDL 导入 Workday Studio 环境。然后，使用“Create and Copy SOAP Literal”或“Open SOAP wrapped request in Web Service Tester”菜单创建有效的 SOAP 请求。
Your IP address failed to authenticate.	您的 IP 地址无权访问 Workday 环境。请与您的系统管理员联系。

Workday Studio 中的多重身份验证

在 Workday Studio 中启用多重身份验证

关于此任务

您可以配置 Workday Studio，使其在连接到租户时使用 OAuth 2.0 授权。为此，您必须注册 API 客户端。该客户端的用户如已在租户中注册 Workday 多重身份验证，则在从 Workday Studio 连接到该租户时，必须使用第二种身份验证方法。他们无法使用基本身份验证。

过程

1. 在您的租户中设置多重身份验证，并指定哪些账户必须使用多重身份验证。

请参见步骤：使用身份验证器应用设置多重身份验证。

请参见步骤：使用 Duo Security 设置多重身份验证。

请参见步骤：使用通过电子邮件发送的一次性密码设置多重身份验证。

请参见步骤：使用一次性短信密码设置多重身份验证。

- 注册启用了 OAuth 2.0 的 API 客户端。
配置以下设置：

选项	值
Client Grant Type	Authorization Grant Code
Support Proof Key for Code Exchange (PKCE)	已启用
Access Token Type	Bearer
Redirection URI	输入有效的 HTTPS URL。
范围	在“Custom Objects”列表中，为您将访问的每项 Web 服务选择功能区域。 此外，选择以下功能区域： • 实施 • 集成 • System • Tenant Non-Configurable 注：如果您忽略了任何相关的 Web 服务，则身份验证将会失败。
Include Workday Owned Scope	已启用

注：请勿选中“Enforce 60 Minute Access Token Expiry”复选框。否则，在该有效期之后，OAuth 2.0 身份验证会被禁用。OAuth 2.0 身份验证还受限于在租户级别配置或者为单个账户配置的常规 Workday 会话超时值。

请参见[注册 API 客户端](#)。

- 复制“注册 API 客户端”任务提供的“客户端编号”“授权端点”“令牌端点”。
 - 从 Workday Studio 的“首选项”菜单中，导航至“Workday > 连接”。
 - 添加租户连接并选择“OAuth 2.0 凭据”选项卡。
 - 粘贴客户端 ID、授权端点和令牌端点。
 - 单击“Get Access Token”。
- 如果您收到内部服务器错误消息，最可能的原因是客户端 ID 不正确。
- 输入租户的用户名和密码。
- 如果您已注册 Workday 多重身份验证，系统会提示您提供第二种身份验证方法。
- 测试连接。
 - 单击“Apply and Close”。

relatedlinks

参考信息

[2021R2 What's New Post: Multifactor Authentication in Workday Studio](#)

集成配置

将集成系统用户账户指定到集成

开始之前

- 将集成部署到 Workday。
- 安全性： “Integration”功能区域中的“Integration Security”域。

关于此任务

Workday Studio 集成需要通过集成系统用户 (ISU) 账户来执行身份验证和 Web 服务调用。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Assign Integration System User”。
5. 单击“Integration System Users”。
6. 从右侧窗格中选择一个集成系统用户。

结果

您可以使用集成系统用户身份启动集成。

下一步做什么

启动您的集成。

概念：集成系统安全性

在 Workday 中，您可以加入安全组，以便在您的租户中执行特定任务。在 Workday Studio 中，当您连接到 Workday 租户时，“Test Connections”窗口会显示 3 个安全组：

- Deploy
- Launch
- 业务流程

Workday Studio 要求您至少是其中 1 个安全组的成员，才能运行集成。

relatedlinks

概念

[概念：可配置的安全性](#)

集成启动

通过 Workday Studio 启动集成

开始之前

- 将集成部署到 Workday。
- 安全性： “Integration”功能区域中的“Integration Event”域。

关于此任务

您可以通过 Workday Studio，在 Workday 租户中启动集成。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Launch Integration”。
5. 单击“启动”。

结果

集成将在您的 Workday 租户中启动。在 Workday Studio 中，“Process Monitor”选项卡会显示集成的状态。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

复制启动 URL

开始之前

将集成部署到 Workday。

关于此任务

您可以通过 Workday Studio，在 Workday 租户中复制集成的启动 URL。您可以使用这些 URL，从具有基本身份验证权限、基于 REST 的第三方客户端启动集成。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Copy Launch URL”。
5. 在“Launch URL”窗口中，选中启动 URL，然后按 Ctrl + C。

结果

Workday Studio 会将集成的启动 URL 复制到剪贴板。

下一步做什么

通过具有基本身份验证权限、基于 REST 的第三方客户端，在 REST 请求中使用该启动 URL。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

创建启动配置

开始之前

- 将集成部署到 Workday。
- 安全性：“Integration”功能区域中的“Integration Event”域。

关于此任务

您可以通过 Workday Studio，在 Workday 租户中启动集成。您可以为集成创建启动配置。

过程

1. 选择“Run”> “Run Configuration”。
2. 在左侧导航窗格中，右键单击“集成”。
3. 单击“New”。
4. 在右侧导航窗格中，请参考以下信息：

选项	描述
名称	启动配置的名称。
Launch In	启动集成的目标 Workday 环境。
Collection	要在集成中启动的集合。
项目	要在集成中启动的项目。
Workday-In	集成中“Workday-In”传输的名称。
Launch Parameters	集成的启动参数。

5. 单击“Apply”。
6. 单击“运行”。

结果

Workday Studio 会使用新的启动配置详细信息来运行集成。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

编辑启动配置

开始之前

- 将集成部署到 Workday。
- 安全性：“Integration”功能区域中的“Integration Event”域。

关于此任务

您可以通过 Workday Studio，在 Workday 租户中启动集成。您可以编辑集成的启动配置。

过程

1. 选择“Run” > “Run Configuration”。
2. 在左侧导航窗格中，展开“集成”节点。
3. 选择一个启动配置节点。
4. 在右侧导航窗格中，请参考以下信息：

选项	描述
名称	启动配置的名称。
Launch In	启动集成的目标 Workday 环境。
Collection	要在集成中启动的集合。
项目	要在集成中启动的项目。
Workday-In	集成中“Workday-In”传输的名称。
Launch Parameters	集成的启动参数。

5. 单击“Apply”。
6. 单击“运行”。

结果

Workday Studio 会使用新的启动配置详细信息来运行集成。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

集成监控和问题排查

流程监视器

概念：集成事件编号

Workday 会为所有集成事件指定一个编号，这使您能够跟踪集成事件从启动到完成的整个过程。从 Workday Studio 启动集成时，集成事件编号将显示在“Process Monitor”视图的“Id”列中。

参考信息：流程监视器工具栏按钮

按钮	描述
刷新	刷新“Process Monitor”视图。
Delete the Process	从“Process Monitor”视图中删除流程。
Copy the Process ID to the Clipboard	将集成事件编号复制到剪贴板。
Cancel the Process	取消流程。

按钮	描述
View Consolidated Report	在“Consolidated Report Viewer”中打开合并的报告。
View Log File	在“View Log File”窗口中显示流程的日志文件。
View Request Message	在“View Request Message”窗口中显示流程的请求消息。
Import Consolidated Report	打开“Import Consolidated Report”窗口。
Import Log File	打开“Import Log File”窗口。
Import Request Message	打开“Import Request Message”窗口。
Enable Auto Refresh	为“Process Monitor”视图启用自动刷新。
Workday: Process Monitor	在“Workday”选项卡上打开 Workday“Process Monitor”。
Workday	打开“Workday”视图并显示集成的集成服务器 ESB 调用。
View Menu	显示“Process Monitor”视图的工具栏选项。

合并的报告查看器

加载有关集成的合并报告

开始之前

- 将集成部署到 Workday。
- 安全性：“Integration”功能区域中的“Integration Event”域。

关于此任务

在 Workday Studio 中，“Consolidated Report Viewer”选项卡显示与集成中每个步骤的用时、内存使用情况和调用次数相关的性能信息。您可以利用合并的报告分析集成流程。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Launch Integration”。
5. 单击“启动”。
6. 在“Process Monitor”视图中，右键单击您的集成。
7. 选择“View Consolidated Report”。

结果

您的合并报告将显示在“Consolidated Report Viewer”选项卡中。

relatedlinks

任务

[将集成部署到 Workday](#) on page 254

[通过 Workday Studio 启动集成](#) on page 260

将合并的报告保存到您的工作区

开始之前

- 将集成部署到 Workday。
- 安全性：“Integration”功能区域中的“Integration Event”域。

关于此任务

您可以利用合并的报告分析集成流程。您可以将合并的报告保存到工作区。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Launch Integration”。
5. 单击“启动”。
6. 在“Process Monitor”视图中，右键单击您的集成。
7. 选择“View Consolidated Report”。
8. 在“Consolidated Report Viewer”选项卡中，右键单击合并的报告。
9. 选择“Save As...”。
10. 在“Save Consolidated Report”窗口中，为合并的报告选择父级文件夹。
11. 单击“Finish”。

结果

Workday Studio 会将合并的报告保存到您的工作区。

relatedlinks

任务

[将集成部署到 Workday](#) on page 254

[通过 Workday Studio 启动集成](#) on page 260

使用合并的报告查看 **Workday** 集成事件详细信息

开始之前

- 将集成部署到 Workday。
- 安全性：“Integration”功能区域中的“Integration Event”域。

关于此任务

您可以利用合并的报告分析集成流程。您可以使用 Workday Studio 中合并的报告查看 Workday 集成事件详细信息。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Launch Integration”。

- 单击“启动”。
- 在“Process Monitor”视图中，右键单击您的集成。
- 选择“View Consolidated Report”。
- 在“Consolidated Report Viewer”选项卡上，选择“View in Workday”。
- 在“Workday”选项卡上，登录您的 Workday 租户。完成此任务时，请参考以下信息：

选项	描述
Username	Workday 租户的用户名凭据。
Password	租户的密码凭据。

结果

您的 Workday 集成事件详细信息将显示在 Workday Studio 中的“Workday”选项卡上。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[通过 Workday Studio 启动集成 on page 260](#)

使用合并的报告查看集成事件的请求消息

开始之前

- 将集成部署到 Workday。
- 安全性：“Integration”功能区域中的“Integration Event”域。

关于此任务

您可以利用合并的报告分析集成流程。您可以使用 Workday Studio 中的合并报告，来查看集成事件的请求消息。

过程

- 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
- 选择并展开一个集合节点。
- 右键单击一个集成节点。
- 选择“Launch Integration”。
- 单击“启动”。
- 在“Process Monitor”视图中，右键单击您的集成。
- 选择“View Consolidated Report”。
- 在“Consolidated Report Viewer”选项卡上，选择“View Request Message”。

结果

集成事件的请求消息将显示在“View Request Message”窗口中。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[通过 Workday Studio 启动集成 on page 260](#)

使用合并的报告查看器事件日志监控集成

开始之前

- 将集成部署到 Workday。
- 安全性：“Integration”功能区域中的“Integration Event”域。

关于此任务

您可以利用合并的报告分析集成流程。您可以在 Workday Studio 中，使用“Consolidated Report Viewer”选项卡内的事件日志来监控集成。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Launch Integration”。
5. 单击“启动”。
6. 在“Process Monitor”视图中，右键单击您的集成。
7. 选择“View Consolidated Report”。
8. 在“Consolidated Report Viewer”选项卡上，选择“Events Log”。

结果

集成的事件日志会在“Consolidated Report Viewer”选项卡上显示。

relatedlinks

任务

[将集成部署到 Workday](#) on page 254

[通过 Workday Studio 启动集成](#) on page 260

使用合并的报告查看器配置文件日志监控集成

开始之前

- 将集成部署到 Workday。
- 安全性：“Integration”功能区域中的“Integration Event”域。

关于此任务

您可以利用合并的报告分析集成流程。您可以在 Workday Studio 中，使用“Consolidated Report Viewer”选项卡内的配置文件日志来监控集成。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Launch Integration”。
5. 单击“启动”。
6. 在“Process Monitor”视图中，右键单击您的集成。
7. 选择“View Consolidated Report”。
8. 在“Consolidated Report Viewer”选项卡上，选择“Profile Log”。

结果

集成的配置文件日志会在“Consolidated Report Viewer”选项卡上显示。所用时区为 PST。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[通过 Workday Studio 启动集成 on page 260](#)

重新播放合并报告的程序集执行情况

开始之前

- 将集成部署到 Workday。
- 安全性：“Integration”功能区域中的“Integration Event”域。

关于此任务

您可以利用合并的报告分析集成流程。您可以使用 Workday Studio 中的合并报告为集成重新播放程序集执行情况。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Launch Integration”。
5. 单击“启动”。
6. 在“Process Monitor”视图中，右键单击您的集成。
7. 选择“View Consolidated Report”。
8. 在“Consolidated Report Viewer”选项卡上，选择“Replay”。

结果

集成的程序集执行情况将在 Assembly Editor 中显示。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[通过 Workday Studio 启动集成 on page 260](#)

将合并的报告审核事件与程序集元素相关联

开始之前

- 将集成部署到 Workday。
- 安全性：“Integration”功能区域中的“Integration Event”域。

关于此任务

您可以利用合并的报告分析集成流程。在 Workday Studio 中，您可以将合并的报告审核事件与程序集元素相关联。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。

2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Launch Integration”。
5. 单击“启动”。
6. 在“Process Monitor”视图中，右键单击您的集成。
7. 选择“View Consolidated Report”。
8. 在“Consolidated Report Viewer”选项卡上，选择“Link with editor”。

结果

Workday Studio 会将合并的报告审核事件与集成中的程序集元素相关联。当您在“Consolidated Report Viewer”选项卡中选择一个节点时，Workday Studio 会在程序集图上突出显示相应的程序集元素。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[通过 Workday Studio 启动集成 on page 260](#)

在程序集图上显示合并的报告注释

开始之前

- 将集成部署到 Workday。
- 安全性：“Integration”功能区域中的“Integration Event”域。

关于此任务

您可以利用合并的报告分析集成流程。在 Workday Studio 中，您可以在程序集图上显示合并的报告注释。

过程

1. 在“Cloud Explorer”视图中，选择并展开一个云服务器节点。
2. 选择并展开一个集合节点。
3. 右键单击一个集成节点。
4. 选择“Launch Integration”。
5. 单击“启动”。
6. 在“Process Monitor”视图中，右键单击您的集成。
7. 选择“View Consolidated Report”。
8. 在“Consolidated Report Viewer”选项卡上，选择“Annotate”。

结果

Workday Studio 会在 Assembly Editor 中的程序集图上显示合并的报告注释。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[通过 Workday Studio 启动集成 on page 260](#)

概念：合并的报告查看器

在 Workday Studio 中，“Consolidated Report Viewer”选项卡显示与集成中每个步骤的用时、内存使用情况和调用次数相关的性能信息。“Consolidated Report Viewer”选项卡还会汇总 Workday Web 服务调用，并突出显示集成错误。

您可以将合并的报告链接到程序集。您还可以在程序集图中的每个程序集元素上显示合并的报告注释，从而发现集成中的问题和错误。

“合并报告查看器”选项卡显示在默认的 Workday Studio 透视面板组中。如果您将其关闭，可以通过选择“窗口”>“显示视图”>“合并报告查看器”来重新打开它。

您可以从“合并报告查看器”选项卡中加载合并报告，也可以从所选集成流程的“流程监视器”视图中加载合并报告。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[通过 Workday Studio 启动集成 on page 260](#)

参考信息：合并的报告错误消息

错误消息	示例	描述
java.lang.Exception : E2 上的 Win : XSLT 作业失败	com.capeclear.mediation.MediationException: Custom step id=WaitJobComplete, ref=WaitJobCompleteBean - Error while invoking bean, reason: Win on E2: XSLT job failed. The Job with referenceId <referenceid> has failed as it exceeded the maximum number of retry attempts, 1 Root Cause: java.lang.Exception: Win on E2: XSLT job failed.	当您的作业已超过重试最大次数时显示。
HttpAuthenticationRequiredException Unauthorized	com.capeclear.mediation.MediationException: RetrieveStep id=null encountered a problem sending data to endpoint <endpoint>, reason:HttpAuthenticationRequiredException: Unauthorized.	当您的权限不足，无法访问附件时显示。
com.workday.esb.blobitory.Blobitory 户 wdom 未处于活动状态	com.capeclear.mediation.MediationException: application=bpelEnterpriseInterfaceBuilder <version-number> - Failed to store resource in collection eib_xslt_collection in workspace wdoms.	租户配置错误。请联系支持部门和技术运营部门，确保租户处于有效状态。
java.lang.OutOfMemoryError: Java 堆空间不足。	com.capeclear.mediation.MediationException: Custom step id=WaitJobComplete,	当您的集成中包含编写质量差的 XSLT 时显示。

错误消息	示例	描述
	<pre> ref=WaitJobCompleteBean - Error while invoking bean, reason: Win on E2: XSLT job failed. Integration Failed. com.capeclear.mediation.MediationException: Xslt step id=transformEIB failed to apply stylesheet mctx:vars/eib.xslt. </pre>	
java.xml.transform.TransformConfig	<pre> com.capeclear.mediation.MediationException: Custom step id=WaitJobComplete, ref=WaitJobCompleteBean - Error while invoking bean, reason: Win on E2: XSLT job failed. Integration Failed. com.capeclear.mediation.MediationException: Xslt step id=transformEIB stylesheet mctx:vars/ eib.xslt located at mctx:vars/eib.xslt is invalid or missing, reason: javax.xml.transform.TransformerConfigurationException: Failed to compile stylesheet. 1 error detected. Error reported by XML parser org.xml.sax.SAXParseException; systemId: mctx:vars/ eib.xslt; lineNumber: 960; columnNumber: 5; The markup in the document following the root element must be well-formed. Root Cause: javax.xml.transform.TransformerConfigurationException: Failed to compile stylesheet. 1 error detected. </pre>	当您的集成中包含无效的 XSLT 时显示。
com.workday.services.server.capec	<pre> com.capeclear.mediation.MediationException: application=bpelACHTransformOutbound-21.0.06.267 - Custom step id=aggregateElement, ref=DocumentElementProcessor - Error while invoking bean, reason: </pre>	当您实施无效的 XSLT 处理器来创建 SoapElementProcessor Transformer 实例。

错误消息	示例	描述
	<pre>document processor step error Root Cause: com.workday.services.server.capeclear.bean.dep.ProcessingExc error transforming SOAP request element, reason:error transforming request element, reason:javax.xml.transform.TransformerException: org.w3c.dom.DOMException: HIERARCHY_REQUEST_ERR: An attempt was made to insert a node where it is not permitted.</pre>	
java.io.FileNotFoundException	<pre>com.capeclear.mediation.MediationException: application=<application- name> - Http Out transport id=retrieveSchemaFromOMS encountered error sending to endpoint <endpoint> Root Cause: java.io.FileNotFoundException: <keystore-file- location> (No such file or directory).</pre>	当 Workday FSB 服务器中缺少密钥库文件时显示。
org.jaxen.UnresolvableException	<pre>com.capeclear.mediation.MediationException: Error eval step id=ValidatePaygroup expression=props['myprop.empl'] = vars['input.picof'].xpath('/ picof:Employee/ picof:Status/ picof:Pay_Group') failed to evaluate. Reason: unable to resolve property: vars['input.picof'].xpath('/ picof:Employee/ picof:Status/ picof:Pay_Group') Root Cause: org.jaxen.UnresolvableException: Cannot resolve namespace prefix 'picof'.</pre>	当程序集中缺少命名空间前缀定义时显示。
BadRequestException	<pre>com.capeclear.mediation.MediationException: Workday Out transport id=CallGetPos encountered problem sending to endpoint</pre>	当集成生成无效的 SOAP 请求消息时显示。

错误消息	示例	描述
	<pre>'https://impl-cc.workday.com/ccx/service/mycompany/Staffing/v17' with operation 'Get_Positions_Request'. Details: Code: SOAP- ENV:Client.validationError, Reason: Validation error occurred. Invalid ID value. '?' is not a valid ID value for type = 'Position_ID' Root Cause: BadRequestException : Error code : unknown.</pre>	
java.net.SocketTimeoutException	<pre>com.capeclear.mediation.MediationException: SMTP Transport error in email out transport id=emFailureRecords with endpoint=mailto: reason:com.capeclear.capecconnect.transport.TransportException Transport messaging error: Unable to send message : Exception reading response Root Cause: java.net.SocketTimeoutException: Read timed out.</pre>	当您的集成中包含无效的 SMTP 详细信息时显示。

relatedlinks

任务

[将集成部署到 Workday on page 254](#)

[通过 Workday Studio 启动集成 on page 260](#)

AUnit 程序集测试

为程序集项目创建 AUnit 测试框架

关于此任务

AUnit 是一个 Workday Studio 测试框架，用于为程序集开发和运行单元测试。AUnit 测试框架使你能够为程序集构建测试用例。

过程

1. 选择“文件” > “新建” > “Workday 程序集项目”。
2. 在“项目名称”字段中，输入项目的名称。
项目名称不能包含空格。
3. 在“Target Runtime”字段中，选择“Workday Runtime”。
4. 在“Configurations”字段中，选择“Assembly (with Java support)”。

5. 单击“完成”。
6. 选择“文件”>“新建”>“其他”>“类”。
7. 在“新建 Java 类”菜单中，请参考以下信息：

选项	描述
Package	输入封装名称。
名称	为新的 Java 类输入名称。
超类	单击“浏览”，然后选择 <code>com.workday.aunit.AssemblyTestCase</code> 。

8. 单击“完成”。
9. 在“Project Explorer”视图中，右键单击项目中的“src”文件夹。
10. 选择“Add JUnit Library”。

结果

Workday Studio 会在 Java 编辑器中显示您的 AUnit 测试框架。

下一步做什么

为您的程序集运行单元测试。

概念：AUnit 测试

AUnit 是一个 Workday Studio 测试框架，可用于为程序集开发和运行单元测试。您可使用在程序集项目中创建的测试框架，通过 Java 开发 AUnit 测试。AUnit 测试框架支持以下功能：

- 操作
- 注释
- 断言

您可使用这些功能为程序集构建测试用例。

示例：

```
package com.workday.training.aunit;

import java.io.InputStream;

import com.workday.aunit.AssemblyTestCase;
import com.workday.aunit.actions.Action;
import com.workday.aunit.actions.StandardAction;
import com.workday.aunit.annotations.AssemblyTest;
import com.workday.aunit.annotations.AssertAfter;
import com.workday.aunit.annotations.AtComponent;
import com.workday.aunit.annotations.UnitTest;

@AssemblyTest(project = "AUnitTrainer", displayLabel = "Citizens of the
Cloud")
public class SimpleMediationTest extends AssemblyTestCase {

    @UnitTest(startComponent = "Simple-Mediation")
    public void testSimpleMediation() throws Exception {
        setMessagePart(0, "test/simple-input.xml", "text/xml");
    }

    // Using the @AssertAfter annotation and the assertTrue assertion.    //
    @AssertAfter(id = "Simple-Mediation", step="transform")
    public void verifySimpleMediation() throws Exception {
```

```
        assertTrue(compare(
            getTestResourceInputStream("test/simple-expected.xml"), "text/
xml",
            (InputStream) getMediationContext().getMessage().getMessagePart(0,
InputStream.class),
            "text/xml", Comparator.dom));
    }

    @AtComponent(id="Next-Thing")
    public Action handleAfterMediation() throws Exception {
        return new StandardAction(Action.Type.terminate);
    }
}
```

注释

所有 AUnit 测试都必须包含 @AssemblyTest 注释此注释可标识您的 AUnit 测试，并使您能够将该测试与您的程序集项目相关联。示例：

```
@AssemblyTest(project = "AUnitTrainer")
```

测试方法中的注释定义了一项单元测试在程序集处理链中的起点。示例：

```
@UnitTest(startComponent = "Simple-Mediation")
```

断言

断言用于定义布尔值测试条件。如果断言失败，AUnit 将中止程序集执行，并显示测试结果为失败。如果发生意外的异常，AUnit 将中止程序集执行，并将测试结果显示为错误。

示例：以下示例使用 @AssertAfter 注释和 assertTrue 声明，以在中介执行后检查根消息部分是否包含所需的 XML。

```
@AssertAfter(id = "Simple-Mediation", step="transform")
public void verifySimpleMediation() throws Exception {
    assertTrue(compare(
        getTestResourceInputStream("test/simple-expected.xml"), "text/xml",
        (InputStream) getMediationContext().getMessage().getMessagePart(0,
InputStream.class),
        "text/xml", Comparator.dom));
}
```

操作

操作定义了程序集执行到达某个构成或退出点时 AUnit 的行为。示例：

```
@AtComponent(id="Next-Thing")
public Action
    handleAfterMediation() throws Exception {
        return new StandardAction(Action.Type.terminate); }
```

参考信息：AUnit 测试注释

注释	描述	示例
@AssemblyTest	将测试标识为 AUnit 测试，并将其与程序集关联。	@AssemblyTest(project = "AUnitTrainer", displayLabel = "Citizens of the Cloud")
@UnitTest	标识单个测试用例。	@UnitTest(startComponent = "Simple-Mediation")

注释	描述	示例
		<pre> public void testSimpleMediation() throws Exception { setMessagePart(0, "test/simple- input.xml", "text/xml"); } </pre>
@AssertBefore	在执行已命名的程序集构成或步骤之前，执行断言。	<pre> @AssertBefore(id="Store- Output") public void verifySummaryDoc() throws Exception { assertEquals("10", evaluateExpression("parts[0].xpath summary/ wd:WorkerSummary)')"));); } </pre>
@AssertAfter	在执行已命名的程序集构成或步骤后执行断言。	<pre> @AssertAfter(id = "Simple-Mediation", step="transform") public void verifySimpleMediation() throws Exception { assertTrue(compare(getTestResource simple-expected.xml"), "text/xml", (InputStream) getMediationContext().getMessage() InputStream.class), "text/xml", Comparator.dom)); } @AssertAfter(id="Process", step="transform") public void verifySummary() throws Exception { assertEquals("Ace Developer", evaluateExpression("parts[0].xpath wd:WorkerSummary/ wd:Name')"));); assertEquals("2010-07-27-07:00", evaluateExpression("parts[0].xpath wd:WorkerSummary/ wd:HireDate')"));); } </pre>
@AssertAfterCompletion	在拼装完成后执行断言。	<pre> @AssertAfterCompletion public void verifyErrorHandling() throws Exception { assertEquals(3, evaluateExpression("props['error.c } </pre>
@AtComponent	定义测试在何处拦截程序集以执行验证，并定义命名组件的测试行为。	<pre> @AtComponent(id="Get- Workers") </pre>

注释	描述	示例
		<pre>public Action handleGetWorkersInvocation() throws Exception { setMessagePart(0, "test/fromworkdayin/ workdayin_workers_response.xml", "text/xml"); return new StandardAction(Action.Type.mock); }</pre>

参考信息：AUnit 测试操作

操作	描述	示例
continue_to_next	继续对程序集中的下一个元素进行 AUnit 测试。	<pre>return new StandardAction(Action.Type.continu</pre>
invoke	调用由程序集的出站传输指定的端点。	<pre>return new StandardAction(Action.Type.invoke)</pre>
mock	通过指定包含响应消息的输入文件来模拟出站传输返回的响应消息。	<pre>return new StandardAction(Action.Type.mock);</pre>
mock_exception	通过指定此类的实例来模拟出站传输器引发的异常： com.capeclear.capeconnect.transport.exceptions.RequestException(msg)	<pre>new StandardAction(Action.Type.mock_ex new transport.exceptions.RequestException(" message here..."));</pre>
terminate	终止测试。	<pre>return new StandardAction(Action.Type.termina</pre>

Web 服务测试

使用 WSDL 文件测试 Web 服务

关于此任务

您可以使用 Workday Studio 的 Web Service Tester，通过执行 WSDL 文件中描述的任何操作来发送示例请求消息。

过程

1. 在 Workday Studio 工具栏上，单击“Test Web Service in Tester”。
2. 完成此任务时，请参考以下信息：

选项	描述
File	显示所选 WSDL 文件的位置。
Workspace	从您的工作区中选择一个 WSDL 文件。
外部	从外部位置选择一个 WSDL 文件。
URL	从 URL 中选择一个 WSDL 文件。

选项	描述
Initialize response endpoint and authentication details for Sandbox environment	指定： - 测试器的“Location”字段是否预填充 Web 服务端点。 - WS-Security 用户名令牌是否预填充您的沙盒用户登录详细信息。 此选项默认处于选中状态。
完成	选择 WSDL 文件后，单击此选项。或者，在未指定文件的情况下单击此选项，以打开 Web Service Tester，然后输入您选择的位置和消息。

- 从您的工作区、外部位置或 URL 中选择一个 WSDL 文件。单击“Finish”。
- Workday Studio 会打开 Web Service Tester。完成此任务时，请参考以下信息：

选项	描述
地点	Web 服务端点。
PortType	Web 服务及其关联的绑定公开的 WSDL 操作。
运算	WSDL 文件中的操作。在您选择其中一项操作后，Workday Studio 会在“Request”窗格中生成示例消息。
SOAP action	HTTP 请求中的 SOAPAction 标头。
Request	Workday Studio 发送到 Web 服务的请求消息。
回复	Workday Studio 从 Web 服务接收的响应消息。

- 根据需要在“Request”窗格中修改 XML。要添加附件，请单击“Add Attachment”，然后单击“Edit”，以指定文件位置以及附件的名称和多用途互联网邮件扩展（MIME 类型）。
- 单击“Send Request”，以将请求消息发送到位于“Location”字段中指定的 URL 的 Web 服务。Workday Studio 会在“Response”窗格中显示响应。

使用 XSD 文件测试 Web 服务

关于此任务

您可以使用 Workday Studio 的 Web Service Tester，通过 XSD 文件中描述的任意元素来发送示例请求消息。

过程

- 在 Workday Studio 工具栏上，单击“Test Web Service in Tester”。
- 完成此任务时，请参考以下信息：

选项	描述
File	显示所选 XSD 文件的位置。
Workspace	从您的工作区中选择一个 XSD 文件。
外部	从外部位置选择一个 XSD 文件。
URL	从 URL 中选择一个 XSD 文件。
Initialize response endpoint and authentication details for Sandbox environment	指定：

选项	描述
	- 测试器的“Location”字段是否预填充 Web 服务端点。 - WS-Security 用户名令牌是否预填充您的沙盒用户登录详细信息。 此选项默认处于选中状态。
完成	选择 XSD 文件后，单击此选项。或者，在未指定文件的情况下单击此选项，以打开 Web Service Tester，然后输入您选择的位置和消息。

- 从您的工作区、外部位置或 URL 中选择一个 XSD 文件。单击“Finish”。
- Workday Studio 会打开 Web Service Tester。完成此任务时，请参考以下信息：

选项	描述
地点	Web 服务端点。由于 XSD 中没有地点值，您必须手动输入地点。
Elements	XSD 文件中的元素定义。在您选择其中一个元素定义后，Workday Studio 会在“Request”窗格中生成示例消息。

- 根据需要在“Request”窗格中修改 XML。要添加附件，请单击“Add Attachment”，然后单击“Edit”，以指定文件位置以及附件的名称和多用途互联网邮件扩展（MIME 类型）。
- 单击“Send Request”，以将请求消息发送到位于“Location”字段中指定的 URL 的 Web 服务。Workday Studio 会在“Response”窗格中显示响应。

通过 Schema Explorer 测试单个 Web 服务

开始之前

在“Schema Explorer”中显示 Workday Web 服务 WSDL 或 XSD。

关于此任务

您可以使用 Workday Studio Web Service Tester，从“Schema Explorer”创建示例 SOAP 或 XML 请求消息，从而测试单个 Web 服务。

过程

- 要为特定 Web 服务生成示例请求消息，请在“Schema Explorer”视图中右键单击 WSDL 或 XSD 元素。选择“Open request in Web Service Tester”或“Open SOAP wrapped request in Web Service Tester”。
- Workday Studio 将打开“WS Tester”视图。“Target”版块显示 Web 服务端点的位置。
- 从“Content-Type”下拉列表中，选择消息的媒体类型和子类型：

选项	描述
文本/纯文本； charset=utf-8	指定纯文本，不包含格式设置命令或指令。
text/xml; charset=utf-8	指定 XML 文本。
application/octet-stream	指定一个无法识别的类型。示例：如果您要在消息中附加二进制文件，请选择此选项。

- 根据需要在“Request”窗格中修改 XML。要添加附件，请单击“Add Attachment”，然后单击“Edit”，以指定文件位置以及附件的名称和多用途互联网邮件扩展（MIME 类型）。

5. 单击“Send Request”，以将请求消息发送到位于“Location”字段中指定的 URL 的 Web 服务。Workday Studio 会在“Response”窗格中显示响应。

relatedlinks

任务

[显示 Workday Web 服务 WSDL on page 213](#)

使用 Request Builder Wizard 测试 Get 操作和请求消息

开始之前

在“Schema Explorer”中显示 Workday Web 服务 WSDL 或 XSD。

关于此任务

您可以使用“Workday Web Service Request Builder Wizard”来简化对多个 Get 操作和请求消息的测试。对于符合条件的操作和请求消息，其图标左上角会显示一个箭头符号。

过程

1. 在“Schema Explorer”视图中，右键单击相关的 Get 节点，然后选择“Workday Web Service Builder Wizard”。
2. 选择您是否想执行以下操作：
 - 使用参考编号检索特定数据。
 - 通过 Web 服务检索所有可用记录。
3. 如果您已选择使用参考编号，请输入该编号。
4. 选择要包括的响应组。
5. 选择您是否想执行以下操作：
 - 将示例请求复制到剪贴板。
 - 在新的“WS Tester”视图中打开示例请求。
 - 替换当前“WS Tester”视图的内容。
6. 单击“Finish”。

配置 Web Service Tester 身份验证

开始之前

创建一项 Web 服务测试。

关于此任务

您可以在 Workday Studio 的 Web Service Tester 中配置多个安全选项。使用“身份验证”选项，您可以指定是将 WS-Security 或基本身份验证应用于 SOAP 请求消息，还是调用内部 cc-authenticate 服务。

过程

1. 在 WS 测试人员 视图中，单击“发送选项”。
2. 选中“Authentication”复选框。完成此任务时，请参考以下信息：

选项	描述
WS-Security UsernameToken	让您能够指定用于 Web 服务安全的用户名和密码，而无需选择 Web 服务安全配置文件。Workday Studio 会在其生成的请求消息的

选项	描述
	SOAP 标头版块插入一个 XML 片段，该片段中包含您指定的用户名和密码。
身份验证服务	Web 服务测试人员调用内部 cc-authenticate 服务 - 使用您指定的用户名和密码登录。如果该用户名和密码在服务器上有效，Workday 会返回一个会话票证，Web 服务测试人员会在发送请求时将该票证添加到 SOAP 标头中。服务器的处理流程包括一个拦截器，用于检查提供的会话票证是否有效。 默认情况下，系统会连接到 cc-authenticate 服务只需创建一次并可重复使用。如果您要停止然后再启动服务器，请选择“在每次发送前重新连接到 cc-authenticate 服务”，以便为每个请求创建新的连接。
Basic Authentication (pre-emptive)	让您能够指定用于抢占式 HTTP 基本身份验证的用户名和密码。

配置 Web Service Tester X.509 身份验证

开始之前

创建 Web 服务测试。为集成系统用户配置一个用户名，该用户名与密钥库别名相对应，但没有密钥库别名。@tenant-name 后缀将相应的基于用户的 Web 服务组指定给集成系统用户。

关于此任务

您可以在 Workday Studio 的 Web Service Tester 中配置多个安全选项。通过“X.509 Authentication”选项，您可以选择与集成系统用户及关联的私钥和公钥对相对应的密钥存储库别名。

过程

1. 在“WS Tester”视图中，单击“Send Options”。
2. 选中“X.509 Authentication”复选框。
3. 从“Username”下拉列表选择一个用户名。每个用户名都与一个密钥存储库别名相关联。
4. (可选。) 单击“Manage Keys”以管理现有 X.509 密钥或生成新的密钥对。在每个会话中，Workday Studio 都会请求一次密钥存储库密码。完成此任务时，请参考以下信息：

选项	描述
Generate X.509 key pair	使您能够生成具有关联数字证书的 X.509 密钥对。
Remove key from the key store	移除选定密钥对。
View the public key (certificate)	显示选定密钥对的公钥。
Import key pair	使您能够执行以下任一操作： • 从 Java 密钥存储库导入私钥和 X.509 证书。 • 使用 OpenSSL PEM 格式导入私钥和 X.509 证书。
Rename alias	使您能够对选定密钥对的密钥存储库别名进行重命名。

配置 Web Service Tester 高级 Web 服务安全性

开始之前

创建一项 Web 服务测试。

关于此任务

您可以在 Workday Studio 的 Web Service Tester 中配置多个安全选项。通过“Advanced WS-Security Configuration”选项，Web Service Tester 能够读取一个安全配置文件，该文件为请求指定一系列安全步骤，为响应指定另外一系列的安全步骤。它将安全步骤应用于传出的请求消息，并从传入的响应消息中移除安全编码。服务器会在接收安全编码时移除安全编码，并在响应安全编码时添加安全编码。

过程

1. 在“WS Tester”视图中，单击“Send Options”。
2. 选中“Advanced WS-Security Configuration”复选框。
3. 提供来自“Workspace”或“External”位置的配置文件 URL。
4. (可选。) 如果安全配置文件中的值 (例如密钥存储位置) 使用的是相对位置，请提供配置基础 URL。Workday Studio 会将此 URL 附加到相对位置。
5. (可选。) 选中“Re-load the security configuration information before each send”复选框，以便为每个请求重新加载文件。如果您更改了配置文件，此选项非常有用。

概念：Web 服务测试

通过 Workday Studio Web Service Tester，您可以根据 Web 服务描述 (WSDL) 或架构定义 (XSD) 创建示例 SOAP 或 XML 请求消息。该工具还会显示 Web 服务返回的任何响应消息。该工具支持 HTTP 和 HTTPS。

下表汇总了启动 Web Service Tester 的 3 种主要方法：

方法	注释
在 Workday Studio 菜单栏中启动。	允许您从工作区、URL 或外部文件位置选择 WSDL 或 XSD 文件进行测试。使用此方法时，您还可以在不引用任何特定文件的情况下打开 Web Service Tester。
从“Schema Explorer”视图中启动。	允许您以非 SOAP 格式或 SOAP 信封封装格式为特定 WSDL 操作或架构元素生成 XML 请求。
使用“Workday Web Service Request Builder”向导来启动。	简化特定 Get 操作和 Get 请求消息的测试。

参考信息：Web Service Tester 工具栏按钮

常规

按钮	描述
Send Request	调用 Web 服务。响应消息显示在“Response”窗格中。
Send Options	打开“Send Options”对话框，供您配置安全选项。
Abort Request	中止当前请求。

按钮	描述
Reload the WSDL	重新加载驱动 Web Service Tester 的 WSDL。如果 Web 服务发生变更，则此选项很有用。
Choose the XML font via the preferences	使您能够为“Request”窗格和“Response”窗格中显示的 XML 设置字体属性。
Request/Response History	提供对先前已发送的请求消息列表的访问权限，使您能够选择并重新发送这些消息。
Request Only	仅显示“Request”窗格。
Response Only	仅显示“Response”窗格。
Horizontal Layout	水平排列布局。
Vertical Layout	垂直排列布局。
Help	显示 Web Service Tester 的上下文相关帮助。

“Request”窗格

按钮	描述
Save Request	将请求消息另存为 XML 文件。
Load Request	加载请求消息。
Format Request	对请求消息应用标准缩进。文本可以超出面板宽度。
Restore original or last sent request	对于未发送的消息，会将文本恢复为其未设置格式和未封装的原始状态。重设任何手动做出的编辑。对于已发送的消息，会将文本恢复到上次发送操作之前的状态。
添加附件	使您能够添加或移除文件附件。要向请求消息添加附件，请单击“Edit”，从工作区中或外部文件位置选择一个文件，然后指定其名称和多用途互联网邮件扩展（MIME 类型）。 要从请求消息中移除附件，请单击“Remove”。 要保存响应消息中包含的附件，请单击“Response”窗格下方的“Open”。
Increase Font	增大“Request”窗格中的文本字体大小。
Decrease Font	缩小“Request”窗格中的文本字体大小。

“Response”窗格

按钮	描述
Save Response	将响应消息另存为 XML 文件。
Load Response	加载响应消息。
Format Response	对响应消息应用标准缩进。文本可以超出面板宽度。

按钮	描述
Wrap Response	应用缩进，同时还会设置面板宽度边界内的文本格式。
Restore Response	对于响应消息，会将文本恢复为其未设置格式和未封装的原始状态。对于示例响应，则从文本窗格中移除已加载的文件。撤消任何手动做出的编辑。
Sample Response	为根据 WSDL 生成的 SOAP 请求消息生成示例响应。
Increase Font	增大“Response”窗格中的文本字体大小。
Decrease Font	缩小“Response”窗格中的文本字体大小。

程序集调试

调试程序集

开始之前

- 创建一个程序集项目。
- 安全性：“Integration”功能区域中的“Integration Debug”域。

关于此任务

通过 Workday Studio，您可以检测和调试程序集中的错误。您可以调试已部署到 Workday 服务器环境的程序集。如果某个程序集包含了引用静态文件的构成或步骤，请在调试之前确保引用的文件路径中没有空格。

过程

1. 在“Project Explorer”视图中，右键单击一个程序集项目。
2. 选择“Deploy to Workday”。
3. 在“Deploy to Workday”窗口中，请参考以下信息：

选项	描述
Workday Environments	选择用于部署集成的目标 Workday 环境。
Collection being deployed	选择一个集合。

4. 单击“Finish”。
5. 在“Cloud Explorer”视图中，选择并展开您将程序集项目部署到的目标 Workday 环境的云服务器节点。
6. 选择并展开您部署到 Workday 的集合。
7. 右键单击集合中的一个集成节点。
8. 选择“Debug Integration”。在“Debug Integration”窗口中，请参考以下信息：

选项	描述
Launch In	启动调试会话的目标 Workday 环境。
Collection	您要调试的集合的名称。
项目	您要调试的项目的名称。
Workday-In	您要调试的项目中“Workday-In”传输的名称。
Launch Parameters	您要调试的项目的启动参数。

选项	描述
	要使用备选启动参数进行调试，请单击“Select Literal Values”，然后从 CRF 返回的 Workday 编号中进行选择。 如果您知道要用作启动参数的具体 Workday 编号，请单击“Edit Literal Value”，然后在“Value”字段中输入该值。

9. 单击“调试”。

注：如果在调试运行时开始启动时遇到问题，请考虑调整“会话重新连接重试次数”值和“最大启动等待迭代次数”值。

Session Reconnect Retries（会话重新连接重试次数）：用于确定 Studio 尝试将调试会话连接到调试运行时的次数。“最大启动等待迭代次数”决定了 Studio 检查调试运行时是否已准备好接收请求的次数。请尝试以 10 为增量来增加它们。

可以在“调试配置”对话框中为现有启动配置调整这些值。

要调整它们以用于将来的启动配置，请选择“窗口 > 首选项”>> “Workday > 程序集调试”>> “调试运行时”。

结果

调试会话的结果将显示在“控制台”视图中。

概念：Assembly Debugger

Workday Studio Assembly Debugger 是一种在 Workday 服务器环境中调试程序集的工具。使用 Assembly Debugger，您可以执行以下操作：

- 检查程序集消息运行时上下文。
- 暂停程序集线程。
- 测试 MVEL 表达式。

如果程序集包含了引用静态文件的构成或步骤，请在调试之前确保引用的文件路径中没有空格。

注：您不能在线租户中调试程序集。请使用非上线环境进行调试。

设置断点

断点可让您在程序集中的特定点处暂停调试过程。要在程序集元素上设置断点，请右键单击一个元素，选择“Toggle Assembly Breakpoint”。在 Assembly Editor 主视图中，该元素的右上角会显示一个蓝色圆圈，用以指示新的断点。

Workday Studio 管理

Workday Studio 首选项

概念：Assembly Editor 首选项

您可以使用“组合件编辑器 首选项”页面，指定组合件关系图的首选项。要导航到首选项页面，请选择“窗口”>“首选项”>“Workday”>“程序集编辑器”。选择以下节点之一，指定相关的程序集编辑器首选项：

- 颜色
- Fonts

- Migration
- Modules

颜色

通过“Diagram Color Preferences”页面，您可为程序集图中的构成指定颜色首选项。

设置	描述
背景颜色	指定程序集构成的背景颜色。
Line Color	指定程序集构成的线条颜色。
文本颜色	指定程序集构成的字体颜色。
Step Background Color	指定程序集步骤的背景颜色。
Header Background Color	指定程序集构成标头的背景颜色。

字体

通过“Diagram Font Preferences”页面，您可为程序集图中的构成指定字体首选项。

设置	描述
默认值	为程序集图中构成的编号值指定标准字体类型、样式和大小。
步骤	为程序集图中步骤的编号值指定标准字体类型、样式和大小。
Conditional Tooltip Header	为程序集图中显示的工具提示标头文本指定标准字体类型、样式和大小。
Conditional Tooltip Content	为程序集图中显示的工具提示内容指定标准字体类型、样式和大小。

迁移

在“Migration”页面上，您可以为使用较早版本的 Workday Studio 创建的旧版程序集项目启用或禁用迁移。

启用迁移后，当您打开一个 assembly.xml 文件时，Workday Studio 会自动将该文件的程序集配置迁移到最新版本的 Workday Studio。

要关闭自动迁移功能，请先取消选中“Enable legacy assembly migration in the Assembly Editor”复选框，然后再打开或保存 assembly.xml 文件。

模块

在开始开发组合件模块模板之前，请在您的工作区或外部位置（例如 Workday Studio 安装中的文件夹）创建一个文件夹，用于存储组合件模块。

程序集编辑器的程序集工具选项板中的“模块库”仅显示从模块文件夹中选择加载的程序集模块，这些模块在“窗口”>“首选项”>“Workday”>“程序集编辑器”>“模块页面”中指定。单击“浏览工作区”或“浏览文件系统”，找到模块文件夹并指定此文件夹的路径。Workday Studio 会在“Modules”表格中显示指定文件夹中的所有程序集模块 XMI（XML 元数据交换）文件。选中要在“Module Library”中加载的每个程序集模块旁的复选框。Workday Studio 会执行模板验证，并且仅加载有效的 XMI 文件。

在设计时，“Assembly Module”向导会提示您将模块 XMI 文件保存到工作区内的文件夹。如果愿意，您可以稍后将程序集模块 XMI 文件从工作区文件夹复制到您在此处指定的模块文件夹（如果该文件夹不同于工作区文件夹）。

概念：连接首选项

在“连接 首选项”页面中，您可以设置集成云连接凭据。要导航到“连接”首选项页面，请选择“窗口” > “首选项” > “Workday” > “连接”。您可以为 沙盒 环境和 上线 环境输入凭据。您还可以添加到其他环境的连接，以及向您的连接添加组。

概念：部署首选项

在“Deployment”首选项页面中，您可以指定在部署到 Workday 环境时是否在 CLAR 中包括项目源代码。要包括，请选中“Include source code in deployed CLAR”复选框。

如果不想在 CLAR 中包括源代码，请取消选中该复选框。

如果您选择“Include source code in deployed CLAR”选项，则可以将源代码导入到工作区中的项目。为此，请在“Cloud Explorer”中右键单击集合，然后选择“Import Source”。

概念：HTTPS 首选项

在“HTTPS”首选项窗格中，您可以指定是否在 Workday Studio 中启用信任库证书管理。如果您选中“Enable truststore certificate management in Studio”复选框，则当 Workday Studio 尝试通过 HTTPS 连接 Workday 时，系统会显示一条消息，提示您信任用于沙盒环境或上线环境的服务器证书。您信任的证书显示在“Truststore Certificates”窗格中，您可以在其中重新加载、移除所有证书或删除选定的证书。要打开“Truststore Certificates”窗格，请展开左侧导航窗格中的“Workday > HTTPS”链接。

概念：Ivy 首选项

Workday Studio 提供 Ivy 插件，让您能够使用 Apache Ivy 来管理项目依赖关系。要使 Ivid 插件能够解析资源，您可以指定要在项目中使用的设置和属性文件。

您可以在 Workday > Ivid 的“首选项”页面上配置以下设置：

- Ivid 设置 URL：单击“浏览...”，然后选择“ivysettings.xml”文件或指定一个 URL。设置文件定义了 Ivy 引擎在解析资源时可以加载的变量。
- Ivy properties file：单击“Browse...”，然后选择设置文件可引用的 .properties 文件。属性文件包含 Ivy 以变量形式加载的属性，其格式如下：

```
name=value
```

选中“Load ivy.properties file from home directory if present”复选框，可自动从您的主目录加载 ivy.properties 文件。

概念：链接首选项

“链接”首选项页面可指定工作区中的链接资源是否使用具有绝对路径的路径变量。要导航至“链接”首选项页面，请选择“窗口” > “首选项” > “Workday” > “Linking”。“将路径变量用于链接资源的绝对路径”设置会自动启用。要禁用此设置，请取消选中相应复选框。

概念：MVEL 验证首选项

在“MVEL Validation”首选项页面中，您可以配置 MVEL 代码验证。“Enable MVEL Validation”设置会自动启用。要禁用此设置，请取消选中相应复选框。

概念：封装集成首选项

“Packaged Integrations”首选项页面仅供内部 Workday Studio 开发者用于封装集成部署。如需有关封装集成支持方面的更多详细信息，内部 Workday Studio 开发者可参阅内部 Wiki。

概念：运行时首选项

Workday Studio 会自动检查 Workday Runtime 更新。如果有任何更新可用，Studio 会在后台下载这些更新，并在安装时通知您。您可以在“窗口”>“首选项”>“Workday”>“Runtime”页面上配置下载、安装和通知首选项。

概念：服务器首选项

在“Server”首选项页面上，您可以调整“Undeploy Workday collections when they are removed from the Server”设置。系统会自动禁用此设置，这意味着，如果某个集合已部署到所选 Workday 环境，并且您已将该集合从“Configured”列表中移除，则 Workday Studio 不会取消部署该集合。要启用此设置，请选中此复选框。当您在“Deploy to Workday”向导中单击“Finish”时，Workday Studio 将取消部署您工作区中不再位于“Configured”列表中的任何集合。

概念：初始屏幕登录首选项

启动时，Workday Studio 会显示一个可选的初始屏幕，您可在其中输入 Workday 环境的详细连接信息。或者，您可以根据自己的偏好，稍后再在“Connection Details”中输入您的详细信息。若要在启动时关闭初始屏幕，请选中“Don't show the login on startup”复选框，然后单击“Apply”以保存更改。

概念：状态栏首选项

通过“Status Bar”首选项页面，您可以指定要在位于 Workday Studio 工作区右下角的“Status Bar”历史记录列表中存储的状态栏条目的最大数量。在“Max Status Bar history items”字段中，输入要存储的最大条目数。

概念：更新首选项

Workday Studio 支持位于低带宽区域的集成商通过 Workday 内容分发网络 (CDN) 进行更新。您可以在“窗口 > 首选项”>“Workday > 安装/更新”页面中启用对 Workday CDN 的访问。

注：如果要使用 CDN，则需要在端口 443 上进行不受限的出站访问。

概念：XPath Explorer 首选项

利用“XPath Explorer”首选项页面，您可以指定在为“Evaluate XPath to XML fragment”选项生成 XML 片段时要使用的命名空间-前缀映射。

要添加映射，请单击“New...”。此时会显示“Add Namespace Prefix Mapping”字段，您可以在该字段中为所需的映射输入前缀值和命名空间值。

要移除不需要的映射，请将其选中并单击“Remove”。要对映射重新排序，请选择一个映射，然后根据需要单击“Up”或“Down”。

参考信息：程序集调试运行时首选项

设置	描述
Hostname	用于启动调试会话的主机名。
HTTP Port	将调试会话连接到 Workday 的端口号。
Session Connect Retries	Workday Studio 尝试将调试会话连接到 Workday 的次数。
Max Startup Wait Iterations	Workday Studio 检查 Workday 是否已准备好接收请求的次数。
Accept All Certificates	指定 Workday Studio 是否接受未签名的证书。
Enable HTTPS Proxy Settings for Debugging	为调试会话启用 HTTPS 代理服务器设置。
Copy Proxy Settings from Eclipse	指定是否从 Eclipse 复制调试会话的代理设置。

设置	描述
Proxy Host	代理服务器主机名。
Proxy Port	代理服务器侦听的端口。
Proxy User Name	代理服务器的用户名。
Proxy Password	代理服务器的密码。
Wait for preceding BP steps timeout, e.g., Retrieval (seconds)	在恢复调试会话之前要经过的超时时限，由 Workday Studio 分配给相关业务流程集成。

参考信息：程序集调试视图首选项

设置	描述
Auto format root part of XML	如果您希望 Workday Studio 自动对消息根部分显示的 XML 进行格式设置，请选中此复选框。
Bring the Assembly Debug View to the front when a thread suspends	如果您希望在调试过程中，当某个线程暂停时，Workday Studio 将“Assembly Debug”视图置于前台，请选中此复选框。
Bring the Exception tab to the front when a thread suspends on an error handler	如果您希望在调试过程中，当某个线程暂停时，Workday Studio 将“Assembly Debug”视图中的“Exception”选项卡置于前台，请选中此复选框。
Automatically retrieve root parts with lengths less than (KB)	指定自动下载阈值，即“Assembly Debug”视图可自动显示的消息根部分的文件大小上限。 若要查看超出自动下载阈值的消息根部分，您必须下载消息，或将其作为文档保存在工作区或文件系统中。为此，请使用“Assembly Debug”视图工具栏按钮。
Truncate messages downloaded to the Assembly Debug View when lengths exceed (KB)	下载后，如果消息在“Assembly Debug”视图中显示，则超出指定文件大小的消息会被截断。

参考信息：Cloud Explorer 首选项

设置	描述
Ask for confirmation when deleting a collection	如果已启用此功能，系统会先提示您确认选择，然后您才能从云中删除集合。
Show annotations in the Cloud Explorer	如果已启用此功能，注释将显示在“Cloud Explorer”中的节点上。
Show Workday connector content	如果启用此功能，“Cloud Explorer”会显示 workday-in 传输的内容，例如启动参数、属性和映射。

参考信息：合并的报告查看器首选项

配置文件日志列	描述
Calls	在集成事件期间，构成收到的调用次数。
Avg Time	构成的平均处理时间。

配置文件日志列	描述
Max Time	构成的最长处理时间。
Min Time	构成的最短处理时间。
Std Deviation Time	与构成平均处理时间之间的标准偏差。
Total Time	构成的总处理时间。
Avg Req Size	构成的平均请求大小。
Max Req Size	构成的最大请求大小。
Min Req Size	构成的最小请求大小。
Std Deviation Req Size	与构成平均请求大小之间的标准偏差。
Total Req Size	构成的总请求大小。
Avg Resp Size	构成的平均响应大小。
Max Resp Size	构成的最大响应大小。
Min Resp Size	构成的最小响应大小。
Std Deviation Resp Size	与构成的平均响应大小的标准偏差。
Total Resp Size	构成的总响应大小。
Max Memory	构成的最大内存用量。
Min Memory	构成的最小内存用量。
Avg Memory	构成的平均内存用量。
Available Memory	构成的可用内存用量。

参考信息：Java 项目设置首选项

设置	描述
Enforce compatibility with only the current workbench default JRE	默认情况下，Workday Studio 生成的任何 Java 代码均兼容任何已安装的 JRE。但通过选中此复选框，您可以强制要求 Java Web 服务项目仅兼容特定 JRE（即默认的工作台 JRE）。
Respect export settings for classpath entries in referenced projects	若选中此复选框，可在导出 WSAR 或部署 Web 服务项目时仅包含从引用的项目中导出的资源。

参考信息：通知首选项

设置	描述
Enable notifications	Workday Studio 会自动启用此设置。取消选中此复选框会禁止在 Workday Studio 屏幕上显示通知。
Fade out delay	输入一个以毫秒为单位的值，以指定通知在淡出之前持续显示的时长。

参考信息：流程监视器首选项

设置	描述
Enable auto refresh	自动刷新“Process Monitor”视图的功能默认处于启用状态。要禁用此功能，请取消选中该复选框。
Refresh interval in seconds	输入一个值（以秒为单位），指定自动刷新的间隔时间。使用右侧的滚动按钮可增大或减小值。
Number of last active processes to refresh	输入要刷新的有效流程的数量。刷新流程是将当前流程和之前的流程按时间倒序排列。
Show error dialog when auto refresh fails	允许在自动刷新功能失败时显示错误消息。Workday Studio 会自动启用此功能。要禁用此功能，请取消选中该复选框。
Show view after launching an integration	当您启动集成时，此功能会在前台显示“Process Monitor”。要禁用此功能，请取消选中该复选框。

参考信息：Web Service Tester 首选项

设置	描述
Request and response XML display	指定显示请求和响应 XML 数据时使用的字号和样式。更改字体详细信息时，已打开的编辑器会立即更新。
Do not display messages larger than:	指定要在 Web Service Tester 的“Response”面板中显示的消息大小上限。标准值为 1024 KB。任何大于指定值的消息都将另存为附件。您可在本地保存附件。

Workday Studio 更新

更新 Workday Studio

关于此任务

所有客户使用的 Workday 应用程序版本只有 1 个，同样，Workday Studio 也只有 1 个有效发行版。您必须安装最新版本的 Workday Studio，才能与沙盒环境或上线环境进行交互。

过程

1. 如有可用更新，Workday Studio 会在您于启动时或启动后输入租户凭据时向您发出通知。确认您要更新，然后按照“Update Workday Studio”向导中的步骤操作。

注：请勿独立于 Workday Studio 更新时间表来更新底层 Eclipse 平台，否则可能会导致集成失败。

2. 如果您选择不立即更新，Workday Studio 会在状态栏中显示一个红色箭头作为提醒。在未安装更新的情况下，您仍可在本地开发程序集。但系统会阻止您连接到沙盒或上线租户。若要稍后更新，请选择“Workday” > “Update Workday Studio”。

注：Workday Studio 包含 Workday Runtime 的轻量级版本，便于您测试和调试集成。为确保您访问的运行时始终与上线租户中的运行时相同，Workday Studio 会在启动时检查运行时更新。如有可用更新，系统会在后台进行下载。要调整此行为，请选择“Window” > “Preferences” > “Runtime” > “Automatic Updates”。

Workday 程序集运行时变更

概念：程序集版本

通过程序集版本管理，Workday 企业服务总线 (ESB) 服务器和 Workday Studio 开发团队可以随着时间的推移在 Workday Runtime 编程模型内逐渐完善程序集，而不会对现有用户产生影响。例如，程序集版本管理使 Workday 能够实现以下目的：

- 弃用功能
- 推出新功能
- 修改现有功能的语义

所有程序集都有一个 version 属性。默认情况下，当您新建程序集项目时，Workday Studio 会设置 version 的属性值 assembly 请将 Assassing.xml 文件中的元素更改为最新版本的 Workday 运行时。格式为 year.drop-number，其中 Drop-number 对应于周数。示例：2016.11。

Studio 会在 项目浏览器中的“程序集”文件夹旁边显示此编号。

在现有运行时功能发生变更时，Workday 会确保这些变更向后兼容。这意味着，所有现有程序集将按照其版本号像之前一样运行。但默认情况下，新程序集的行为方式将与当前版本号对应。通常，当 Workday 引入变更时，我们会为程序集步骤或构成定义新的属性。较早版本的程序集没有此属性，因此我们引入了基于版本号的规则，以便为较早的程序集设置默认值。我们还引入了一项规则，可默认为较新的程序集使用不同的值。作为开发者，如果您想在旧版程序集中使用新功能，您有多种选择。您可以：

- 保留旧版本，并接受 Workday Runtime 指定的默认值。
- 保留旧版本，并根据您的需求，为新属性设置显式值。
- 迁移到较新版本，并接受新的默认值。
- 迁移到较新版本，并为新属性设置显式值。

参考信息：架构级别上的版本变更

下表汇总了每次版本更新的架构级别变更。

版本	构成	属性	变更描述
8.0	async-mediation 构成 和 sync-mediation 构成	处理下游错误	默认值更改自 true 至 false 值为 false 表示此组件中的本地错误处理程序仅处理在此组件的步骤中发生的错误。 值为 true 表示此组件中的本地错误处理程序可以处理下游错误。
9.0	write	output-mime-type	默认值更改自 text/plain 至 text/xml，与其他步骤一致。
11.0	xmldiff 步骤	copy-to-output	默认值已更改为 true。 值为 true 表示此步骤会将输出复制到中介消息中。

版本	构成	属性	变更描述
			值为 false 表示该步骤将输出复制到变量。
12.0	http-in 传输	安全	默认值更改自 false 至 null，这表示除非另有说明，否则可通过 HTTP 和 HTTPS 使用该端点。
	http-out 传输协议	error-as-response	默认值更改自 false 至 true。 值为 true 表示当运行时在传输中遇到错误时，它会在当前消息中设置该错误。以前，只能通过 <code>BadRequestException</code> 上 <code>MediationContext</code>。
	local-out、http-out、ftp-out、ftps-out、sftp-out、xmpp-out、custom-out、email-out 传输协议	ignore-dynamic-endpoints	默认值更改自 false 至 true。 值为 true 表示传输将忽略传入邮件标头中的动态 WS-Addressing (WSA) 值。 值为 false 表示传输协议使用 WSA 值来重置其 Endpoint 属性中的值。
13.0	store	schema	默认值更改自 <code>urn:com.workday/bsvc/blob</code> 至 <code>[http://www.w3.org/2005/Atom]</code>。 在向当前的集成事件添加文档时，<code>PutIntegrationMessage</code> 子程序集可直接使用 Avo。
	async-mediation 构成和 sync-mediation 构成	continue-after-error	默认值更改自 <code>recover</code> 至 <code>rewind</code>。 值为 <code>rewind</code> 表示在错误被清除后，系统会从组件中的上一个构成继续进行处理。 上一个默认值 <code>recover</code>，现已弃用。
	本地输出传输	取消设置属性	默认值更改自 false 至 true。

版本	构成	属性	变更描述
			值为 true 表示传输属性在调用子部件号后重置。
	on-error 错误处理程序组件	重新引发错误	默认值更改自 true 至 false。 值为 false 表示运行时将在错误处理程序触发时清除错误。 值为 true 表示错误处理程序必须自行清除错误。
14.0	workday-out-soap	替换为 soap-fault	默认值更改自 false 至 true。 值为 true 表示返回时，传输会将当前消息替换为 SOAP 错误。
	local-in	访问	默认值更改自 public 至 private。 值为 private 表示仅 local-out 在同一程序集中可以调用传输器。 值为 public 表示任何程序集均可调用传输。
	聚合器组件的 message-content-collater 策略	output-mimetype	默认值更改自 application/octet-stream 至 text/plain。 自定义整理者 AggregationCollater 界面仍默认为 application/octet-stream。
18.0	store	immutable	新属性，用于指定是否可以直接修改 Bobs 引用的二进制数据。默认为 true。 值为 true 表示该 bobs 不可变，无法修改。 值为 false 表示该二进制大对象是可变的，可以修改。 注：在 Workday 18 中，服务器运行时会迁移具有旧版本号的现有程序集上的所有 存储步骤，以使用这一新的

版本	构成	属性	变更描述
			默认设置，该设置会使二进制对象 (BLO) 不可变。如果您的程序集需要更新二进制对象，请确保将“不可变”设置为 false。
2016.45 注：在版本 27 以后，Workday 引入了新的命名方案，形式为“”。 year .week	PagedGet	is.paged.get.version	默认值更改自 util.assemblyVersionAsWSVer 至 null。
2019.30	PagedGet	As_Of_Effective_Date As_Of_Entry_DateTime	如果未为这些属性提供值，它们会默认设为当前日期和时间 (PST)。
2020.09	ftp-out、ftps-out、sftp-out		Workday 现在使用 JSCAPE 版本 9.3，支持更广泛的加密密码集合。
2020.09	PrismAnalytics	wd.prism.component.api.howmany	Workday Studio 现可调用 Prism Analytics REST API 版本 2。 构成名称已从 PrismConnector 更改为 PrismAnalytics。

[relatedlinks](#)

[参考信息](#)

[常见问题：用于集成的加密、证书和密码](#)

参考信息：**XML** 级别上的版本变更

下表汇总了每次版本更新的 XML 级别变更。

版本	先前的 XML 元素/属性	替换条目	变更的含义/原因
11.0 (这些变更全部涉及对 Workday-in 传输组件中的集成系统定义所做的变更。)	simple-type-reference	simple-type	正在声明而非引用条目的类型。
	type-reference	class-report-field	新的元素名称更好地体现了元素的含义。
	service	service-reference	不是正在定义条目，而是正在引用条目。移除了 defined-by 属性 (已不再使用) 。
	integration-system/@template	已移除	不再适用。
	param/@required	param/@option	更改了布尔值 required 属性

版本	先前的 XML 元素/属性	替换条目	变更的含义/原因
			option 值为""的属性 required (如果为真)。
	cloud/name	cloud/@name	简洁明了。
	integration/name	in-connector/@id	简洁明了。
	in-connector/id	private-key	private-key
12.0	key-file	private-key	影响 sftp-out 与 ftps-out 传输重命名了属性以更好地反映其含义。
	public-key-properties	private-key-properties	已重命名以更好地反映其含义。
	aggregator/@spawn-aggregated-processing	已移除	此功能已允许聚合器新建任务或线程来处理聚合消息。此功能未在 Workday 集成中使用，也不适合用于公开的 Workday Studio，因此已将其移除。
	store/@compress store/@encrypt	已移除	之前允许对 store 步骤进行精细控制。但是，Workday 关于集成的策略是始终进行压缩和加密，因此它们被移除。
	ftp-out/@tmp-directory sftp-out/@tmp-directory ftps-out/@tmp-directory	已移除	之前，这些传输自行管理自己的临时文件，但后续已进行重构以使用通用的 FileBackedManagedData (FBDM) 功能，因此这些选项已被移除。
14.0	delivery-service/child::*	已移除	传送服务的子元素定义了允许的传输。此类限制不应在设计时定义。这是一项运行时配置任务，因此已将其从设计时开发阶段移除。

参考信息：Saxon XSLT 处理器变更

下表汇总了 Workday 对各种版本的 Saxon 和 XSLT 的支持。请注意，在版本 27 以后，Workday 引入了以下形式的新命名方案：year.week。

Workday 版本	支持的 Saxon 版本	支持的 XSLT 版本	注释
<20	9.1	1.0、(2.0)	为 XSLT 2.0 版提供有限支持。

Workday 版本	支持的 Saxon 版本	支持的 XSLT 版本	注释
20-2017.05	9.4	2.0、(3.0)	为 XSLT 3.0 版提供有限支持。
2017.05-	9.7	3.0	支持 XSLT 流式传输。

注：如果您创建的集合中包含版本号不同的程序集，Workday Studio 会显示警告消息。这是因为不同版本的程序集所需要的 Saxon 版本可能相互冲突，导致在运行时出现类加载错误。最佳做法是为所有集成使用最新程序集版本。

请注意以下事项：

- MVEL 中可用于访问消息部分和变量的 XPath 方法支持 XPath 3.0。
- 该 mctx URL 协议允许 XSL 开发人员向消息部件和变量写入消息，例如：

```
<xsl:result-document href="mctx:vars/var1">...</xsl:result-document>
```

- 尽管 XQuery 不提供直接程序集支持，但您可以添加自定义步骤来调用 XQuery 脚本。
- 在版本为 2022.46 或更早版本的程序集中，仅当 `use.saxon.transformer` 属性设置为 `true` 在中介环境中。在版本为 2022.47 或更高版本的程序集中，如果将该属性设置为 `true` 或（如果不存在）。

注：XSLT 2.0 和 XSLT 3.0 属于强类型。例如，`xs:string` 将在 XPath 表达式中评估“”是否相等。显示以下表达式，其中 `stringParam` 是一个 XSL 参数，会导致 Saxon 运行时错误：

```
$stringParam = true()
```

XSL 编译器在部署时不会遇到此问题，因为它无法提前预测参数的类型。

除非您通过将 XSL 文档中的 `版本` 属性值更改为来选择启用 XSLT 3.0，否则您现有的 XSL 文档和 Workday Studio 集成将不会受此强类型问题的影响。3.0。如果您遇到这些由通过集成部署的 XSLT 导致的运行时错误，则必须执行修复操作。

注：在用于业务表单打印的旧版 XSL 中发现了与 Saxon 升级相关的以下错误：

```
Failed to compile stylesheet.Xerrors detected. Invalid value for @case-order. Value must be one of (lower-first|upper-first).
```

如果您遇到此错误，请检查 XSL 的排序元素是否包含以下属性设置：

```
case-order="#default"
```

此可选项 `case-order` 属性决定了排序时是以大写字母在前还是以小写字母在前。默认为首先列出大写字母。对于 `Case-Order`，Saxon 解析器仅接受以下值：

- `upper-first`
- `lower-first`

要修复此错误，请执行以下操作之一：

- 从排序元素中移除现有的 `case-order` 属性，因为这是一项可选设置。
- 将 `Case-Order` 值明确设置为 `upper-first`。
- 将 `Case-Order` 值明确设置为 `lower-first`。

示例组合件

示例：运行“Hello Cloud”示例程序集

关于此任务

您可以通过 Workday Studio，在 Workday 中部署和启动集成。此示例演示了如何运行基本的示例集成。

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure
- Integration Event

过程

1. 在 Studio 工具栏上，选择“帮助”>“欢迎使用”>“示例”>“Hello Cloud”。
2. 在“Project Names”窗口中，单击“完成”。
3. 在“Project Explorer”视图中，展开“HelloCloud”文件夹，然后双击“Assembly”文件夹。
4. 在“Assembly Editor”工具栏上，单击“Show annotations”。
5. 在“Cloud Explorer”视图中，单击“Connection Details”。
6. 选择“Sandbox”环境，并在“Connections”窗口中输入以下值：

选项	描述
租户	租户的名称。
Username	租户的用户名凭据。
Password	租户的密码凭据。

7. 单击“Apply and Close”。
8. 在“Project Explorer”视图中，右键单击“HelloCloud”文件夹，然后选择“Deploy to Workday”。
9. 在“Deploy to Workday”窗口中，选择“Sandbox”环境。单击“完成”。
10. 在云浏览器视图中，展开以下节点：沙盒环境 > HelloCloudCollection > HelloCloud。
11. 右键单击“StartHere”集成节点，然后选择“启动集成”。
12. 单击“启动”。

结果

集成将在您的 Workday 租户中启动。Workday Studio 中的“Process Monitor”视图会显示集成的状态。

下一步做什么

登录 Workday 并访问“流程监视器”任务，以查看示例程序集的集成事件详细信息。

示例：运行“Hello Workday Web Services”示例程序集

关于此任务

Workday Studio 允许您连接到 Workday Web 服务。此示例演示了 SOAP Web 服务请求消息 Get_Workers_Request。它使用 XSLT 3.0 流式传输来处理请求。

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure
- Integration Event

过程

1. 在 Workday Studio 工具栏上，选择“Help”>“Welcome”>“Samples”>“Hello Workday Web Services”。
2. 在“Project Names”窗口中，单击“Finish”。
3. 在“Project Explorer”视图中，展开“HelloWWS”文件夹，然后双击“Assembly”文件夹。
4. 在“Assembly Editor”工具栏上，单击“Show annotations”。
5. 在“Cloud Explorer”视图中，单击“Connection Details”。
6. 选择“Sandbox”环境，并在“Connections”窗口中输入以下值：

选项	描述
租户	租户的名称。
Username	租户的用户名凭据。
Password	租户的密码凭据。

7. 单击“Apply and Close”。
8. 在“Project Explorer”视图中，右键单击“HelloWWS”文件夹，然后选择“Deploy to Workday”。
9. 在“Deploy to Workday”窗口中，选择“Sandbox”环境，然后单击“Finish”。
10. 在“Cloud Explorer”视图中，展开以下节点：“Sandbox”>“HelloWWSCollection”>“HelloWWS”。
11. 右键单击“HelloWWS_EntryPoint”集成节点，然后选择“Launch Integration”。
12. 单击“启动”。

结果

集成将在您的 Workday 租户中启动。Workday Studio 中的“Process Monitor”视图会显示集成的状态。

下一步做什么

登录 Workday 并访问“流程监视器”任务，以查看示例程序集的集成事件详细信息。

示例：运行常见使用模式示例程序集

关于此任务

此示例演示了一些常见的集成设计模式。其中包含多个独立的处理链，每个处理链说明了一种不同的模式。

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure
- Integration Event

过程

1. 在 Workday Studio 工具栏上，选择“Help”>“Welcome”>“Samples”>“Common Usage Patterns”。
2. 在“Project Names”窗口中，单击“Finish”。
3. 在“Project Explorer”视图中，展开“CommonUsagePatterns”文件夹，然后双击“Assembly”文件夹。
4. 在“Assembly Editor”工具栏上，单击“Show annotations”。
5. 在“Cloud Explorer”视图中，单击“Connection Details”。
6. 选择“Sandbox”环境，并在“Connections”窗口中输入以下值：

选项	描述
租户	租户的名称。
Username	租户的用户名凭据。
Password	租户的密码凭据。

7. 单击“Apply and Close”。
8. 在“Project Explorer”视图中，右键单击“CommonUsagePatterns”文件夹，然后选择“Deploy to Workday”。
9. 在“Deploy to Workday”窗口中，选择“Sandbox”环境，然后单击“Finish”。
10. 在“Cloud Explorer”视图中，展开以下节点：“Sandbox”>“CommonUsagePatterns”>“CommonUsagePatterns”。
11. 右键单击“Conditional-Processing”集成节点，然后选择“Launch Integration”。
12. 单击“启动”。
13. 启动以下各个集成节点：

- Enriching
- Global-Error-Handling
- Local-Error-Handling
- Looping
- Request-Response-Processing-Chain
- Response-Chaining
- Routing
- Scalable-Message-Processing
- Simple-Processing-Chain
- SubRoutines
- 变量

“请求-响应-处理链”和“响应链”集成具有一个额外的启动参数：Employee_ID。单击“Select literal value”进行配置。

结果

集成将在您的 Workday 租户中启动。Workday Studio 中的“Process Monitor”视图会显示集成的状态。

下一步做什么

登录 Workday 并访问“流程监视器”任务，以查看示例程序集的集成事件详细信息。

示例：运行调试示例程序集

关于此任务

通过 Workday Studio，您可以检测和调试集成中的错误。此示例演示了如何调试示例集成。

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure
- Integration Debug
- Integration Event

过程

1. 在 Workday Studio 工具栏上，选择“Help” > “Welcome” > “Samples” > “Assembly Debugger”。
2. 在“Project Names”窗口中，单击“Finish”。
3. 在“Project Explorer”视图中，展开“AssemblyDebug”文件夹，然后双击“Assembly”文件夹。
4. 在 Assembly Editor 中，右键单击以下各个程序集元素，然后选择“Toggle Assembly Breakpoint”：
 - create-get-workers-request
 - store-summary-doc
5. 在“Cloud Explorer”视图中，单击“Connection Details”。
6. 选择“Sandbox”环境，并在“Connections”窗口中输入以下值：

选项	描述
租户	租户的名称。
Username	租户的用户名凭据。
Password	租户的密码凭据。

7. 单击“Apply and Close”。
8. 在“Project Explorer”视图中，右键单击“AssemblyDebug”文件夹，然后选择“Deploy to Workday”。
9. 在“Deploy to Workday”窗口中，选择“Sandbox”环境，然后单击“Finish”。
10. 在“Cloud Explorer”视图中，展开以下节点：“Sandbox” > “AssemblyDebug” > “AssemblyDebug”。
11. 右键单击“Start-Here”集成节点，然后选择“Debug Integration”。
12. 单击“Debug”。
13. 在“Confirm Perspective Switch”窗口中，单击“Yes”。
14. 在“Assembly Debug”视图中，选择“Message Root Part”选项卡，然后单击“Save the message or attachment to a document in the workspace”。
15. 在“Save”窗口中，输入以下值：

选项	描述
Parent Folder	AssemblyDebug
File name	rootpart.xml

16. 在“Debug”视图中，右键单击“Create-Request - create-get-workers-request”断点，然后选择“Resume”。
17. 在“Assembly Debug”视图中，选择“Message Root Part”选项卡，然后单击“Save the message or attachment to a document in the workspace”。

18. 在“Save”窗口中，输入以下值：

选项	描述
Parent Folder	AssemblyDebug
File name	rootpart1.xml

19. 在“Debug”视图中，右键单击“Store - store-summary-doc”断点，然后选择“Terminate”。

结果

Workday Studio 会将 XML 断点文件保存到“AssemblyDebug”项目文件夹中。您可以使用这些文件来排查程序集中的错误。

下一步做什么

打开程序集的 XML 断点文件。

示例：运行 Workday-In 功能示例程序集

关于此任务

在 Workday 中，“workday-in”传输构成表示为一个集成系统。此示例演示了如何在 Workday Studio 程序集中，使用 MVEL 表达式来引用集成系统的服务和启动参数。其中包含 3 个单独的处理链。

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure
- Integration Event

过程

1. 在 Studio 工具栏上，选择“帮助”>“欢迎使用”>“示例”>“Workday-In 功能”。
2. 在“Project Names”窗口中，单击“完成”。
3. 在“Project Explorer”视图中，展开“WorkdayInFeatures”文件夹，然后双击“Assembly”文件夹。
4. 在“Assembly Editor”工具栏上，单击“Show annotations”。
5. 在“Cloud Explorer”视图中，单击“Connection Details”。
6. 选择“Sandbox”环境，并在“Connections”窗口中输入以下值：

选项	描述
租户	租户的名称。
Username	租户的用户名凭据。
Password	租户的密码凭据。

7. 单击“Apply and Close”。
8. 在“Project Explorer”视图中，右键单击“WorkdayInFeatures”文件夹，然后选择“Deploy to Workday”。
9. 在“Deploy to Workday”窗口中，选择“Sandbox”环境，然后单击“Finish”。

10.对于每项单独的集成，请参考以下信息：

集成	注释
AttributesAndMaps	部署后，您可以在 Workday 中调整属性和映射值。 a. 访问“View Integration System”报告并选择 <i>WorkdayInFeatures-attr-maps</i>。 b. 从“ExampleAttrMapService”集成服务的相关操作菜单中，选择“Integration Generic Service > Configure Integration Attributes”或“Integration Generic Service > Configure Integration Maps”。 c. 根据需要调整属性值和映射值。 .
启动参数	在 Studio 的“云资源管理器”视图中，展开以下节点：“沙盒环境”>“WorkdayInFeatures”>“WorkdayInFeatures”。 a. 在“referenceParam”字段中，单击“选择字面值”，然后选择一名员工。 b. Enter 键 password 在“passwordParam”字段中配置密码。
序列生成器	部署后，您可以在 Workday 中配置“序列生成器”服务： a. 访问“View Integration System”报告并选择“<i>WorkdayInFeatures-sequences</i>”。 b. 从“ExampleSeqGeneService”集成服务的相关操作菜单中，选择“Integration Sequence Generator Service”>“Configure Integration Sequence Generators”。 c. 在“增量”字段中，输入 1。 d. 在“格式/语法”字段中，输入 <code>file[Seq].csv</code>。

11.在 云浏览器 视图中，展开以下节点：沙盒 > 环境 WorkdayInFeatures > WorkdayInFeatures。

12.右键单击 Workday-in 传输节点，然后选择“启动集成”。示例：AttributesAndMaps。

13.单击“启动”。

结果

集成将在您的 Workday 租户中启动。Workday Studio 中的“Process Monitor”视图会显示集成的状态。

下一步做什么

登录 Workday 并访问“流程监视器”任务，以查看示例程序集的集成事件详细信息。

示例：运行错误处理示例程序集

关于此任务

此示例演示了在 Workday Studio 中处理错误的几种方法。其中包含 6 个不同的处理链，每个处理链说明了一种不同的模式。

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure
- Integration Event

过程

1. 在 Workday Studio 工具栏上，选择“Help”>“Welcome”>“Samples”>“Error Handling”。
2. 在“Project Names”窗口中，单击“Finish”。
3. 在“Project Explorer”视图中，展开“ErrorHandling101”文件夹，然后双击“Assembly”文件夹。
4. 在“Assembly Editor”工具栏上，单击“Show annotations”。
5. 在“Cloud Explorer”视图中，单击“Connection Details”。
6. 选择“Sandbox”环境，并在“Connections”窗口中输入以下值：

选项	描述
租户	租户的名称。
Username	租户的用户名凭据。
Password	租户的密码凭据。

7. 单击“Apply and Close”。
8. 在“Project Explorer”视图中，右键单击“ErrorHandling”文件夹，然后选择“Deploy to Workday”。
9. 在“Deploy to Workday”窗口中，选择“Sandbox”环境，然后单击“Finish”。
10. 在“Cloud Explorer”视图中，展开以下节点：“Sandbox”>“ErrorHandling”>“ErrorHandling”。
11. 右键单击某个 workday-in 传输节点，然后选择“Launch Integration”。示例：All-Or-Nothing。
12. 单击“启动”。

结果

集成将在您的 Workday 租户中启动。Workday Studio 中的“Process Monitor”视图会显示集成的状态。

下一步做什么

登录 Workday 并访问“流程监视器”任务，以查看示例程序集的集成事件详细信息。

示例：运行 AUnit 示例测试

关于此任务

AUnit 是一个 Workday Studio 测试框架，可用于对程序集执行单元测试。此示例演示了如何运行 AUnit 测试。它包含 8 个独立的处理链。

过程

1. 在 Studio 工具栏上，选择“帮助”>“欢迎”>“示例”>“单元测试”。
2. 在“Project Names”窗口中，单击“完成”。
3. 在“Project Explorer”视图中，展开“AUnit 101”文件夹，然后双击“Assembly”文件夹。
4. 在“Assembly Editor”工具栏上，单击“Show annotations”。
5. 运行“AUnit 101”示例测试。
 - a) 在“Project Explorer”视图中，展开“src”文件夹。
 - b) 右键单击 com.workday.example.aunit 文件夹。
 - c) 选择“Run As”>“AUnit Test”。

结果

Studio 会在“Console”视图中显示示例测试的结果。

下一步做什么

导航到“JUnit”视图，查看详细的结果汇总。

示例：运行 FOPStep 示例程序集

关于此任务

Workday Studio 使您能够在集成输出中生成 PDF 文件。此示例演示了如何生成基本的 PDF 文档。其中包含 3 个单独的处理链。

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure
- Integration Event

过程

1. 在 Studio 工具栏上，选择“帮助”>“欢迎”>“示例”>“FOP 步骤”。
2. 在“Project Names”窗口中，单击“完成”。
3. 在“Project Explorer”视图中，展开“FOPStep”文件夹，然后双击“Assembly”文件夹。
4. 在“Cloud Explorer”视图中，单击“Connection Details”。
5. 选择“Sandbox”环境，并在“Connections”窗口中输入以下值：

选项	描述
租户	租户的名称。
Username	租户的用户名凭据。
Password	租户的密码凭据。

6. 单击“Apply and Close”。
7. 在“Project Explorer”视图中，右键单击“FOPStep”文件夹，然后选择“Deploy to Workday”。
8. 在“Deploy to Workday”窗口中，选择“Sandbox”环境。单击“完成”。
9. 在 云浏览器 视图中，展开以下节点：沙盒 > FOPStep > FOPStep。
10. 右键单击“Simple-Start”集成节点，然后选择“启动集成”。

11. 单击“启动”。
12. 在“Process Monitor”视图中，右键单击该集成，然后选择“Workday: Integration ESB Invocation”。
13. 在“Workday”视图中，输入以下值：

选项	描述
Username	租户的用户名凭据。
Password	租户的密码凭据。

14. 在“View Background Process”窗口中，选择“Output Files”选项卡。
15. 在“Reports and Other Output Files”表格的“File”列中，双击 PDF 文件。
16. 启动“Invoice-Start”和“Kanji-Start”集成节点。

结果

Workday 会显示新生成的 PDF 文件。

下一步做什么

下载并保存 PDF 文件。

示例：运行“Extract Organization Data for a Worker”示例程序集

关于此任务

通过 Workday，您可以使用组织类型将员工划分到组中。此示例演示了如何运行集成，以从 Workday 中提取员工的组织类型和员工编号值。

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure
- Integration Event

过程

1. 在 Studio 工具栏上，选择“帮助”>“欢迎使用”>“示例”>“提取员工的组织数据”。
2. 在“Project Names”窗口中，单击“完成”。
3. 在“Project Explorer”视图中，展开“OrgDataForWorkerForOrgType”文件夹，然后双击“Assembly”文件夹。
4. 在“Assembly Editor”工具栏上，单击“Show annotations”。
5. 在“Cloud Explorer”视图中，单击“Connection Details”。
6. 选择“Sandbox”环境，并在“Connections”窗口中输入以下值：

选项	描述
租户	租户的名称。
Username	租户的用户名凭据。
Password	租户的密码凭据。

7. 单击“Apply and Close”。
8. 在“Project Explorer”视图中，右键单击“OrgDataForWorkerForOrgType”文件夹，然后选择“Deploy to Workday”。

9. 在“Deploy to Workday”窗口中，选择“Sandbox”环境。单击“完成”。
10. 在云浏览器视图中，展开以下节点：沙盒 > 环境“OrgDataForWorker” > “OrgDataForWorkerForOrgType”。
11. 右键单击“StartGetOrgForOrgType”集成节点，然后选择“启动集成”。
12. 在“Launch Integration”窗口中，导航至“Launch Parameters”版块。
13. 在“OrgType”字段中，单击“Select literal values”。
14. 在“Select Class Report Field”窗口中，单击“Next”。
15. 在“CRF”窗口中，选择一个组织类型。完成此任务时，请参考以下信息：

选项	描述
名称	组织类型的名称。
WID	组织类型的 Workday 编号。

16. 单击“Finish”。
17. 在“EmployeeID”字段中，单击“Select literal values”。
18. 在“Select Class Report Field”窗口中，单击“Next”。
19. 在“CRF”窗口中，选择一名员工。完成此任务时，请参考以下信息：

选项	描述
名称	员工姓名。
WID	员工的 Workday 编号。

20. 单击“Finish”。
21. 单击“启动”。
22. 在“Process Monitor”视图中，右键单击您的集成。
23. 选择“Show Details View”。
24. 在“Process Details”视图中，展开“OrgType”和“EmployeeID”启动参数字段。

结果

Workday Studio 会在“Process Details”视图中显示员工的组织类型和员工编号值。

示例：运行 PrismAnalytics 示例程序集

关于此任务

本示例演示了如何使用 PrismAnalytics 构成将 CSV 数据发送到 Workday Prism Analytics。程序集项目中提供了 CSV 源文件。对于本示例，请将这些文件复制到 SFTP 服务器，以供示例程序集检索。

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure
- Integration Event

过程

1. 注册 Prism API 客户端，并复制 PrismAnalytics 构成所需的值。
 - a) 在 Workday 中，访问“Register API Client for Integrations”任务。
 - b) 注册一个 API 客户端，并将 Prism Analytics 作为其范围（功能区域）。选中“Include Workday Owned Scope”复选框。
 - c) 复制客户端 ID 值和客户端密码值。
 - d) 从相关操作菜单中，选择“API Client” > “Manage Refresh Tokens for Integrations”。
 - e) 选择一个要与刷新令牌关联的 Workday 账户。该账户必须有权在 Workday 中创建 Prism 表格。
 - f) 生成刷新令牌并复制值。
2. 获取 Workday REST API 端点。
 - a) 访问“View API Clients”报告。
 - b) 复制 Workday REST API 端点值。删除 https:// 位于开头，并且 /ccx/api/v1/<tenant_name> “”
3. 在 Studio 工具栏上，选择“帮助” > “欢迎使用” > “示例” > “PrismAnalytics”。
4. 在“Project Names”窗口中，单击“Finish”。
5. 在“Project Explorer”视图中，展开“PrismAnalytics”文件夹，然后双击“Assembly”文件夹。
6. 将提供的示例 CSV 文件复制到您的 SFTP 服务器。
 - a) 在 PrismAnalytics 项目文件夹中，展开“ws” > “WSAR-INF” > “PrismAnalytics Sample CSVs”。
 - b) 将以下文件复制到您的 SFTP 服务器：
 - states.csv
 - states2.csv
 - states3.csv
 - states4.csv
 - states5.csv
7. 在“Assembly Editor”工具栏上，单击“Show annotations”。
8. 在“Cloud Explorer”视图中，单击“Connection Details”。
9. 选择“Sandbox”环境，并在“Connections”窗口中输入以下值：

选项	描述
租户	租户的名称。
Username	租户的用户名凭据。
Password	租户的密码凭据。
10. 单击“Apply and Close”。
11. 在“Project Explorer”视图中，右键单击“PrismAnalytics”文件夹，然后选择“Deploy to Workday”。
12. 在“Deploy to Workday”窗口中，选择“Sandbox”环境，然后单击“Finish”。
13. 在“Cloud Explorer”视图中，启动本示例中的一个集成。
 - a) 展开“Sandbox”节点。
 - b) 展开“PrismAnalyticsCollection”集合节点。
 - c) 展开“PrismAnalytics”项目节点。
 - d) 右键单击某个 workday-in 传输节点，然后选择“Launch Integration”。示例：SingleCSV。
14. 单击“启动”。

结果

Workday Prism Analytics 会使用您提供的 CSV 数据创建一个表格。

第三方薪资接口示例

步骤：运行薪资接口示例

关于此任务

在此示例中，一个名为“PayData Extract”的集成会为薪资组提取与一次性付款相关的数据。第二个 Studio 集成会循环显示集成事件文档，并使用 XSLT 3.0 流式传输来处理这些文档。

过程

1. [创建薪资接口安全组](#) on page 308.

创建一个安全组，使其有权访问薪资接口示例所需的域。

2. 访问“Create Integration System User”任务。

为“PayData Extract”集成创建一个集成系统用户。示例：PayData Extract ISU。

从集成系统用户的相关操作菜单中，选择“Security Profile” > “Assign Integration System Security Groups”。在“Integration System Security Group to Assign”提示中，选择“Payroll Interface Security”。

安全性：“Integration”功能区域中的“*Integration Security*”域

3. [创建 PayData Extract 集成系统](#) on page 309.

在 Workday 中创建集成，以提取薪资组的一次性付款数据。

4. [测试 PayData Extract 集成系统](#) on page 310.

5. [部署 Workday Studio PayData 集成](#) on page 311.

部署一个 Studio 集成，该集成将循环显示“PayData Extended”集成事件文档，并使用 XSLT 3.0 流式传输来处理这些文档。

6. 访问“Create Integration System User”任务。

为“PayData”集成创建一个集成系统用户。示例：PayData ISU。

安全性：“Integration”功能区域中的“*Integration Security*”域

7. [为 PayData 集成指定集成系统用户](#) on page 312.

8. [将 PayData Extract 集成与 PayData 集成相关联](#) on page 312.

9. [启动已关联的集成](#) on page 313.

创建薪资接口安全组

开始之前

安全性：“System”功能区域中的“Security Configuration”域

关于此任务

您需要先在 Workday 中创建安全组，然后才能在 Workday Studio 中运行薪资接口示例。

过程

1. 登录您的 Workday 测试环境。
2. 访问“Create Security Group”任务。完成此任务时，请参考以下信息：

选项	描述
Type of Tenanted Security Group (租户设置的安全组类型)	Integration System Security Group (Unconstrained)

选项	描述
名称	Payroll Interface Security

- 访问“查看安全组”报告。
- 从“安全组”提示中，选择“Payroll Interface Security”。
- 从该安全组的相关操作菜单中，选择“安全组” > “Maintain Domain Permissions for Security Group”。
- 从“提供读取权限的域安全策略”提示中,选择以下域：
 - 薪资接口
 - Manage: Organization Integration
 - Worker Data: Public Worker Reports
 - Worker Data: Compensation by Organization

Workday 会提醒您必须激活安全策略变更。

- 访问“Activate Pending Security Policy Changes”任务并输入备注。
- 选中“Confirm”复选框。

结果

Workday 会将您的新安全组添加到每个相关功能区域的域安全策略中。

下一步做什么

创建集成系统用户，以运行“PayData Extract”集成。

创建 PayData Extract 集成系统

开始之前

安全性：“Integration”功能区域中的“Integration Build”域和“Integration Configure”域。

关于此任务

在 Workday 中，创建一个用于提取一次性付款数据的集成。

过程

- 登录您的 Workday 测试环境。
- 访问“Create Integration System”任务。
- 完成“Create Integration System”任务时，请参考以下信息：

选项	描述
System Name	PayData Extract
New Using Template	薪资接口

- 在“Configure Integration Services”页面上，选中以下集成服务对应的“Enabled”复选框：
 - Payroll Interface/PI Change Detection Launch Parameters
 - Payroll Interface/PI Pay Data Section Fields
 - Payroll Interface/PI Filename

Workday 会提醒您必须为某些集成属性赋值，并为集成系统配置序列生成器。

- 在“PayData Extract”集成系统的相关操作菜单中，选择“Workday Account” > “Edit”。
- 在“Workday Account”提示中，输入“PayData Extract”集成系统的集成系统用户名称。
- 在“Edit Account for Integration System”页面上，单击“OK”。

8. 在“PayData Extract”集成系统的相关操作菜单中，选择“Integration System” > “Configure Integration Attributes”。完成此任务时，请参考以下信息：

选项	描述
版本	添加一行并选择值“25”。
Output Document Tags	添加一行，选择“Create Document Tag”，然后在“Document Tag Name”中输入“PICO”。
Primary Payroll Integration	添加一行。
Payroll Vendor	添加一行并选择任意供应商。

9. 在“PayData Extract”集成系统的相关操作菜单中，选择“Integration System” > “Configure Integration Field Attributes”。

10. 在“Field Configuration”版块中，选中以下字段对应的“Include in Output”复选框：

- 代码
- 日期
- Earning or Deduction
- 金额

11. 在“PayData Extract”集成系统的相关操作菜单中，选择“Integration System” > “Configure Integration Sequence Generators”。

12. 在“Configure Integration Sequence Generators”页面上，请参考以下信息：

选项	描述
Increment by	1
Format/Syntax	薪资接口-paydata-[Seq].xml

结果

Workday 会对“PayData Extract”集成系统进行配置。

下一步做什么

测试“PayData Extract”集成系统。

测试 PayData Extract 集成系统

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure
- Integration Event

关于此任务

您需要先在 Workday 中测试薪资接口示例的集成系统，然后才能在 Workday Studio 中运行薪资接口示例。

过程

1. 登录您的 Workday 测试环境。
2. 访问“View Integration System”任务。
3. 在“Integration System”提示中，输入“PayData Extract”。
4. 在“PayData Extract”集成系统的相关操作菜单中，选择“Integration” > “Launch/Schedule”。

5. 在“为集成排定时间表”页面上的“值”列中，输入以下值：

选项	描述
薪资组	选择具有一次性付款的任意薪资组。
薪资期间选择	使用当前日期的薪资期间。
Last Successful Run	选择一个日期，在该日期之后有新的一次性付款产生。

6. 启动集成并刷新，直至其完成。

结果

“PayData Extract”集成系统将在 Workday 中启动。

下一步做什么

将“PayData”集成从 Workday Studio 部署到 Workday。

部署 Workday Studio PayData 集成

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure

关于此任务

Workday Studio PayData 集成会循环访问“PayData Extract”集成事件文档，并使用 XSLT 3.0 流式传输来处理这些文档。您必须先将此集成部署到 Workday，然后才能启动。

过程

1. 在 Workday Studio 工具栏上，选择“Help” > “Welcome” > “Samples” > “Payroll Interface”。
2. 在“Project Names”窗口中，单击“Finish”。
3. 在“Project Explorer”视图中，展开“PayData”文件夹，然后双击“Assembly”文件夹。
4. 在“Assembly Editor”工具栏上，单击“Show annotations”。
5. 在“Cloud Explorer”视图中，单击“Connection Details”。
6. 选择“Sandbox”环境，并在“Connections”窗口中输入以下值：

选项	描述
租户	租户的名称。
Username	租户的用户名凭据。
Password	租户的密码凭据。

7. 单击“Apply and Close”。
8. 在“Project Explorer”视图中，右键单击“PayData”文件夹，然后选择“Deploy to Workday”。
9. 在“Deploy to Workday”窗口中，选择“Sandbox”环境。单击“Finish”。

结果

Workday Studio 会将 PayData 示例集成部署到 Workday。

下一步做什么

创建集成系统用户，以运行 PayData 集成。

为 **PayData** 集成指定集成系统用户

开始之前

安全性：“Integration”功能区域中的“Integration Configure”域

关于此任务

在 Workday 中，为“PayData”集成系统指定集成系统用户。

过程

- 1. 在“Cloud Explorer”视图中，展开以下节点：“Sandbox” > “PayData” > “PayData”。
- 2. 右键单击“StartHere”集成节点。
- 3. 选择“Workday: Integration System”。
- 4. 在“Workday”视图中，请参考以下信息：

选项	描述
Username	Workday 租户的用户名凭据。
Password	租户的密码凭据。

- 5. 在“PayData”集成系统的相关操作菜单中，选择“Workday Account” > “Edit”。
- 6. 在“Workday Account”提示中，输入“PayData”集成系统的集成系统用户名称。
- 7. 在“Edit Account for Integration System”页面上，单击“OK”。

结果

Workday 会为“PayData”集成系统指定集成系统用户。

下一步做什么

将“PayData Extract”集成与“PayData”集成相关联。

将 **PayData Extract** 集成与 **PayData** 集成相关联

开始之前

安全性：“Integration”功能区域中的“Integration Configure”域

关于此任务

在 Workday 中将 PayData Extract 集成与 PayData 集成相关联。

过程

- 1. 登录您的 Workday 测试环境。
- 2. 访问“View Integration System”任务。
- 3. 在“Integration System”提示中，输入“PayData Extract”。
- 4. 在“PayData Extract”集成系统的相关操作菜单中，选择“Business Process” > “Create, Copy, or Link Definition”。
- 5. 单击“OK”新建业务流程定义。

6. 在“Edit Business Process Integration Process Event for PayData Extract (TOP LEVEL)”页面的“Business Process Steps”选项卡中，单击“Add Row”。在新行中，请参考以下信息：

选项	描述
顺序	C
类型	集成

Workday 会警告您配置尚未完成。

7. 单击“Configure Integration System”。
8. 在“Integration”提示中，选择“PayData”。
9. 在“Configure Integration Step”页面的“Integration Attributes”选项卡上，请参考以下信息：

选项	描述
值类型	在运行时确定值
值	可交付文档

结果

Workday 会将“PayData Extract”集成和“PayData”集成关联在一起，以提供所需的输出。

下一步做什么

在 Workday 中启动已关联的“PayData Extract”集成和“PayData”集成。

启动已关联的集成

开始之前

安全性：“Integration”功能区域中的以下域：

- Integration Build
- Integration Configure
- Integration Event

关于此任务

启动现已关联到 PayData 集成的“PayData Extract”集成。

过程

1. 登录您的 Workday 测试环境。
2. 访问“View Integration System”任务。
3. 在“Integration System”提示中，输入“PayData Extract”。
4. 在“PayData Extract”集成系统的相关操作菜单中，选择“Integration” > “Launch/Schedule”。
5. 在“为集成排定时间表”页面上的“值”列中，输入以下值：

选项	描述
薪资组	选择具有一次性付款的任意薪资组。
薪资期间选择	使用当前日期的薪资期间。
Last Successful Run	选择一个日期，在该日期之后有新的一次性付款产生。

6. 启动集成并刷新，直至其完成。

7. 在“Event Documents”版块中，注意 Workday 生成的 CSV 文件。示例：payrollinterface-paydata-1.csv。