

IBM Business Process Manager
Versão 8 Release 5

*Visão Geral do IBM Business Process
Manager*



Manuais PDF e o Centro de Informações

Os manuais PDF são fornecidos como uma conveniência para impressão e leitura off-line. Para obter as informações mais recentes, consulte o centro de informações on-line.

Como um conjunto, os manuais PDF contêm o mesmo conteúdo que o centro de informações. Alguns links nos manuais de PDF foram padronizados para uso nos centros de informações e podem não funcionar corretamente.

A documentação PDF está disponível dentro de um trimestre após um release principal do centro de informações, tal como a Versão 7.0 ou a Versão 7.5.

A documentação PDF será atualizada com menos frequência que o centro de informações, mas com mais frequência do que os Redbooks. Em geral, manuais PDF são atualizados quando mudanças suficientes são acumuladas para o manual.

Índice

Manuais PDF e o Centro de Informações.	iii
---	------------

Capítulo 1. Introdução ao IBM Business Process Manager.

Process Manager.	1
Visão Geral do Produto.	1
Configurações do IBM Business Process Manager	3
Recursos de Configuração do IBM Business Process Manager	4
O Repositório do Process Center.	4
Process Server e Ambientes de Tempo de Execução	5
Ambientes de Criação	6
Ferramentas de Administração	7
Acessibilidade no IBM Business Process Manager	9
Disponibilidade de Idiomas Nacionais no IBM Business Process Manager	9
Visão Geral do Gerenciamento de Processos de Negócios	10
Visão Geral da Modelagem de Processo	11
Desenvolvimento de Processo com o Process Center	12
Aplicativos de Processo: Visão Geral	13
Executando e Depurando Processos	19
Instalando e Gerenciando Aplicativos de Processo	20
Criando, Acessando e Incorporando Serviços	22
Acessando Serviços Externos a um Aplicativo	22
Criando ou Chamando um Serviço da Web.	28
Aprenda Mais sobre os Conceitos Chave	29
Criação de Versões	30
Versão dos Aplicativos de Processo	30
Criando Versão de Módulos e Bibliotecas	31
Módulos e Bibliotecas Associados com Aplicativos de Processo ou Kits de Ferramentas	32
Convenções de Nomenclatura	32
Convenções de Nomenclatura para Implementações do Servidor do Process Center	33
Convenções de Nomenclatura para Implementações do Process Server.	36
Ligações Reconhecidas pela Versão	37
Chamada Dinâmica Reconhecida pela Versão	39
Implementando Aplicativos de Processo com Módulos e Projetos Java	40
Implementando Aplicativos de Processo com Regras de Negócios e Seletores	40
Objetos de Configuração	40
Arquitetura de Implementação	40
Células	40
Servidores	41
Servidores Independentes	41
Clusters	42
Perfis	42
Gerenciadores de implementação	44

Nós	44
Nós Gerenciados	44
Nós não gerenciados	45
Agentes do nó	45
Considerações sobre a Nomenclatura de Perfis, Nós, Servidores, Hosts e Células	45
BPMN 2.0	49
Business Process Definitions (BPDs)	51
Ligações	52
Visão Geral da Ligação de Exportação e Importação	54
Configuração da Exportação e Importação de Ligação.	58
Transformação do Formato de Dados em Importações e Exportações	59
Seletores de Função em Ligações de Exportação	62
Manipulação de Falha	64
Interoperabilidade entre módulos SCA e serviços SCA abertos	69
Tipos de Ligação	72
Selecionando Ligações Apropriadas	72
Ligações SCA.	73
Ligações de Serviço da Web	74
Ligações HTTP	98
Ligações EJB.	106
Ligações EIS.	113
Ligações JMS	118
Ligações JMS Genéricas	127
Ligações JMS doWebSphere MQ	134
Ligações WebSphere MQ	141
Limitações de Ligações	151
Objetos de Negócios	153
Definindo objetos de negócios.	153
Trabalhando com objetos de negócios	154
Objetos de Negócios Especiais.	156
Modo de Análise de Objeto de Negócios	157
Considerações ao Escolher o Modo de Análise do Objeto de Negócios	157
Benefícios de Usar o Modo de Análise Lento Versus Mais Rápido	158
Considerações sobre Migração e Desenvolvimento de Aplicativo	159
Relacionamentos	161
Serviço de Relacionamento	163
Gerenciador de Relacionamento	163
Relacionamentos em Ambientes de Implementação de Rede	164
APIs do Serviço de Relacionamento	164
O Barramento de Serviço Corporativo no IBM Business Process Manager	165
Conectando Serviços Através de um Enterprise Service Bus	165
Infraestrutura do Sistema de Mensagens do Barramento de Serviço Corporativo	166

Hosts do Sistema de Mensagens ou do Destino de Fila	167	Ligações de Dados	214
Provedores JDBC	167	Seletores de Função em Ligações de Exportação	215
Barramento de Integração de Serviço para IBM Business Process Manager	168	Manipulação de Falha	217
Aplicativos de Serviço e Módulos de Serviço	168	Como as Falhas São Manipuladas em Ligações de Exportação	218
Importações e Ligações de Importações	169	Como as Falhas São Manipuladas em Ligações de Importação	220
Exportações e Ligações de Exportações	170	Interoperabilidade entre módulos SCA e serviços SCA abertos	222
Módulos de Mediação	171	Tipos de Ligação	225
Primitivas de Mediação	174	Selecionando Ligações Apropriadas	225
Roteamento Dinâmico	179	Ligações SCA	226
Controle de Política de Mediação de Solicitações de Serviço	180	Ligações de Serviço da Web	227
WebSphere Service Registry and Repository	180	Visão geral das ligações de serviço da Web	227
WebSphere eXtreme Scale	181	Propagação de Cabeçalho SOAP	228
Cientes do Serviço de Mensagens	182	Propagação do Cabeçalho de Transporte	231
		Trabalhando com Ligações de Serviço da Web (JAX-WS)	233
		Anexos em mensagens SOAP	236
		Uso da Ligação de Estilo do Documento WSDL com Mensagens Multipartes	250
Capítulo 2. Aprenda Mais sobre os Conceitos Chave	183	Ligações HTTP	251
Criação de Versões	183	Visão Geral das Ligações HTTP	252
Versão dos Aplicativos de Processo	183	Cabeçalhos HTTP	253
Criando Versão de Módulos e Bibliotecas	184	Ligações de Dados HTTP	257
Módulos e Bibliotecas Associados com Aplicativos de Processo ou Kits de Ferramentas	185	Ligações EJB.	259
Convenções de Nomenclatura	185	Ligações de importação EJB	259
Convenções de Nomenclatura para Implementações do Servidor do Process Center.	186	Ligações de exportação EJB.	261
Convenções de Nomenclatura para Implementações do Process Server	189	Propriedades de ligação EJB	262
Ligações Reconhecidas pela Versão	190	Ligações EIS.	266
Chamada Dinâmica Reconhecida pela Versão	193	Visão Geral das Ligações EIS	266
Implementando Aplicativos de Processo com Módulos e Projetos Java	193	Recursos-chave de Ligações EIS	267
Implementando Aplicativos de Processo com Regras de Negócios e Seletores	193	Propriedades Dinâmicas de Especificação de Interação JCA e de Especificação de Conexão	269
Objetos de Configuração	193	Clientes Externos com Ligações EIS	271
Arquitetura de Implementação	194	Ligações JMS	271
Células	194	Visão Geral de Ligações JMS	272
Servidores	194	Integração de JMS e Adaptadores de Recursos	273
Servidores Independentes	195	Ligações de Importação e Exportação JMS	273
Clusters	195	Cabeçalhos JMS	275
Perfis	196	Esquema de correlação de destino de resposta dinâmica temporária do JMS	276
Gerenciadores de implementação.	197	Clientes Externos	277
Nós	197	Resolução de Problemas de Ligações JMS	278
Nós Gerenciados	197	Exceções de Manipulação	279
Nós não gerenciados	198	Ligações JMS Genéricas	280
Agentes do nó	198	Visão Geral de Ligações JMS Genéricas	280
Considerações sobre a Nomenclatura de Perfis, Nós, Servidores, Hosts e Células	198	Recursos-Chave de Ligações JMS Genéricas.	283
BPMN 2.0	202	Cabeçalhos JMS Genéricos	285
Business Process Definitions (BPDs)	205	Resolução de Problemas de Ligações JMS Genéricas.	285
Ligações	205	Exceções de Manipulação	287
Visão Geral da Ligação de Exportação e Importação	208	Ligações JMS doWebSphere MQ	287
Configuração da Exportação e Importação de Ligação	211	Visão Geral de Ligações JMS do WebSphere MQ.	287
Transformação do Formato de Dados em Importações e Exportações	212	Recursos-chave das Ligações JMS do WebSphere MQ.	290
Manipuladores de Dados	212		

Cabeçalhos JMS	291	Relacionamentos	314
Clientes Externos	292	Serviço de Relacionamento	316
Resolução de Problemas de Ligações JMS do WebSphere MQ	293	Gerenciador de Relacionamento	316
Exceções de Manipulação	294	Relacionamentos em Ambientes de Implementação de Rede	317
Ligações WebSphere MQ	294	APIs do Serviço de Relacionamento	317
Visão Geral de Ligações do WebSphere MQ.	295	O Barramento de Serviço Corporativo no IBM Business Process Manager	318
Recursos-chave de uma Ligação do WebSphere MQ.	297	Conectando Serviços Através de um Enterprise Service Bus	318
Cabeçalhos do WebSphere MQ	299	Infraestrutura do Sistema de Mensagens do Barramento de Serviço Corporativo	319
Incluindo o MQCIH estaticamente em uma ligação do WebSphere MQ	301	Hosts do Sistema de Mensagens ou do Destino de Fila	320
Clientes Externos	302	Armazém de Dados	320
Resolução de Problemas de Ligações WebSphere MQ.	302	Provedores JDBC	320
Exceções de Manipulação	303	Barramento de Integração de Serviço para IBM Business Process Manager	321
Limitações de Ligações	304	Aplicativos de Serviço e Módulos de Serviço Importações e Ligações de Importações.	322
Limitações da Ligação do MQ.	304	Exportações e Ligações de Exportações	323
Limitações das Ligações JMS, JMS do MQ e JMS Genéricas	304	Módulos de Mediação	324
Objetos de Negócios	306	Primitivas de Mediação	327
Definindo objetos de negócios	306	Roteamento Dinâmico	332
Trabalhando com objetos de negócios	307	Controle de Política de Mediação de Solicitações de Serviço	333
Objetos de Negócios Especiais.	309	WebSphere Service Registry and Repository WebSphere eXtreme Scale	333 334
Modo de Análise de Objeto de Negócios	310	Clientes do Serviço de Mensagens	335
Considerações ao Escolher o Modo de Análise do Objeto de Negócios	310		
Benefícios de Usar o Modo de Análise Lento Versus Mais Rápido	311		
Considerações sobre Migração e Desenvolvimento de Aplicativo	312		

Capítulo 1. Introdução ao IBM Business Process Manager

Entenda quais recursos o IBM® Business Process Manager fornece para gerenciamento de processo de negócios e como as várias fases do gerenciamento de processos de negócios, como criar e implementar os aplicativos de processo, se relacionam entre si.

O aplicativo de processo é o contêiner fundamental para processos e seus componentes no IBM Business Process Manager. Designers de processo criam aplicativos de processo nos ambientes de autoria e podem incluir serviços, tarefas e artefatos necessários para suportar a execução.

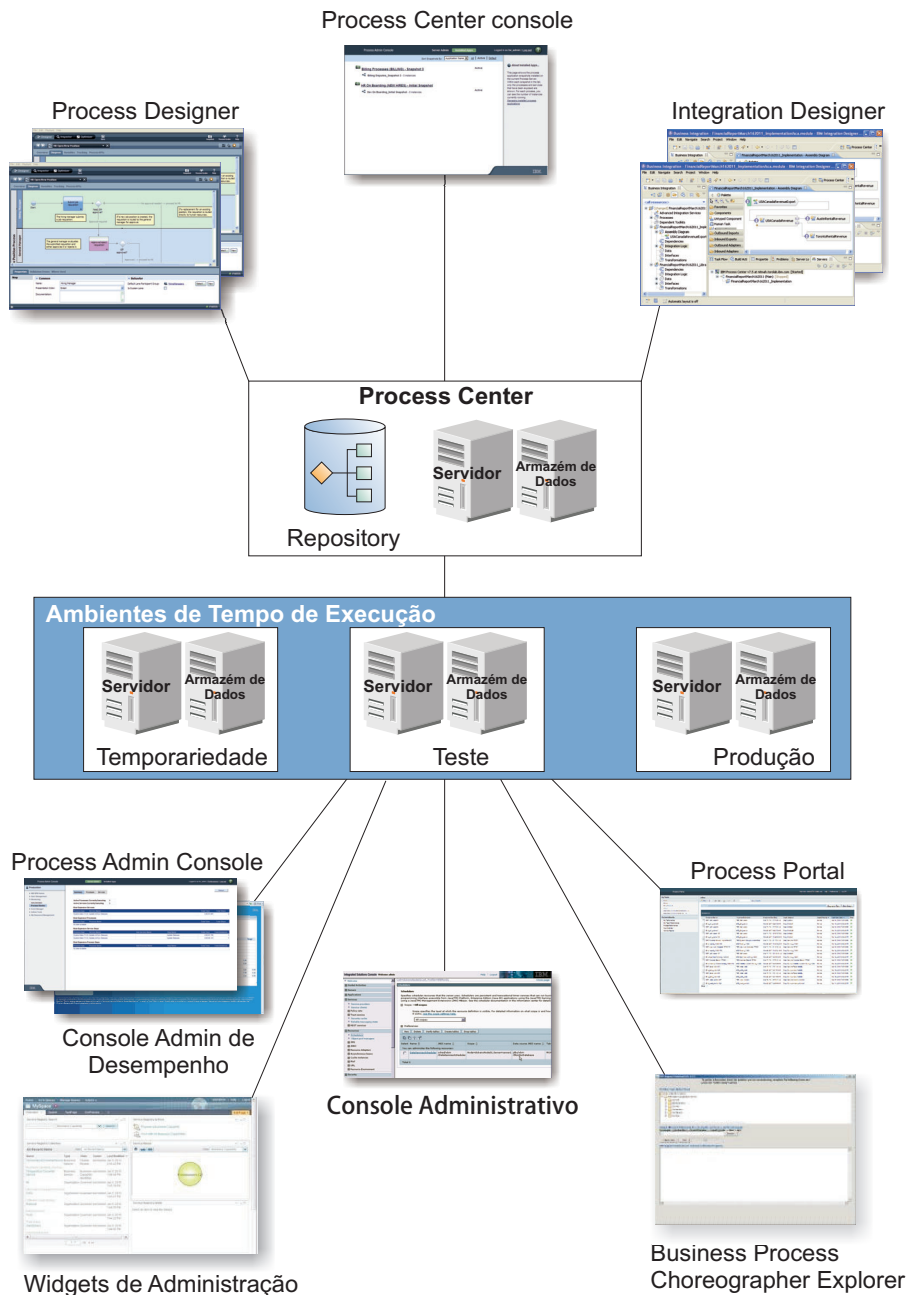
Serviços de Integração Avançados são implementados no IBM Integration Designer e associados a aplicativos de processo. No Process Center, os aplicativos de processo são implementados no Process Server, que é o ambiente de tempo de execução do processo para IBM Business Process Manager.

Da mesma forma, processos automatizados criados no Integration Designer podem usar serviços manuais que foram desenvolvidos no IBM Process Designer.

Visão Geral do Produto

O IBM Business Process Manager é uma plataforma de gerenciamento de processo de negócios abrangente, que fornece visibilidade integral e insight para gerenciar processos de negócios. Ele fornece conjunto de ferramentas e um ambiente de tempo de execução para design, execução, monitoramento e otimização de processo, juntamente com o suporte de integração de sistema de configuração básica. O produto pode ser configurado para suportar vários níveis de complexidade e envolvimento com gerenciamento de processos de negócios.

Os componentes do IBM Business Process Manager fornecem um repositório BPM unificado, conjuntos para autores, administradores e usuários, e uma plataforma de tempo de execução. O produto pode ser configurado para suportar vários níveis de complexidade e envolvimento com gerenciamento de processos de negócios. O diagrama a seguir ilustra uma configuração de IBM Business Process Manager típica:



- Nos ambientes de autoria do IBM Process Designer e do IBM Integration Designer, os desenvolvedores se conectam ao IBM Process Center. A partir de uma dessas ferramentas de desenvolvimento de aplicativo baseadas na GUI, os desenvolvedores podem criar, testar, depurar e implementar aplicativos de negócios. Escolha uma ou outra ferramenta, dependendo do tipo de aplicativo que está sendo desenvolvido. Também pode haver casos em que o uso de ambas as ferramentas traz vantagens significativas.
- Nos ambientes de autoria do Process Designer e do Integration Designer, designers de processos e serviços criam aplicativos de processo implementáveis e kits de ferramentas reutilizáveis. Os aplicativos de processo contêm modelos de processo e implementações de serviço, incluindo os arquivos de suporte necessários. Os aplicativos de processo são armazenados no repositório do Process Center para que possam ser compartilhados.
- O Process Center inclui dois servidores, o servidor Process Center e o servidor Performance Data Warehouse. Esses servidores permitem que desenvolvedores que estejam trabalhando no Process Designer executem seus aplicativos de processo e armazenem dados de desempenho para teste e

reprodução durante os esforços de desenvolvimento. O Performance Data Warehouse recupera dados controlados do Process Server ou do servidor Process Center em intervalos regulares.

- O Process Center também suporta inúmeras funções administrativas. No Process Center Console, os administradores instalam aplicativos de processo que estão prontos para temporariedade, teste ou produção nos servidores de processos. Os administradores também podem gerenciar instâncias em execução de aplicativos de processo em ambientes configurados.
- Instale aplicativos de processo em um servidor de processos para temporariedade, teste e produção. Os ambientes de tempo de execução suportam os processos Business Process Model and Notation (BPMN) 2.0. O IBM Business Process Manager Advanced também suporta processos Business Process Execution Language (BPEL).
- No Process Admin Console e no Performance Admin Console, administradores podem gerenciar e manter todos os servidores de runtime. Use o Process Admin Console para gerenciar o servidor Process Center e os servidores de processos nos ambientes de tempo de execução. Use o Performance Admin Console para identificar gargalos de desempenho e para capturar dados de instrumentação para análise adicional.
- Use o console administrativo para criar e gerenciar objetos, tais como recursos, aplicativos e servidores. Além disso, use o console administrativo para visualizar mensagens do produto.
- Use o Business Space para criar espaços de negócios customizados que forneçam widgets para monitorar ou administrar diferentes aspectos de seu sistema. Por exemplo, é possível monitorar atividades de negócios, serviços e funcionamento do sistema ou administrar políticas de mediação e calendários de negócios. Também é possível criar um espaço de negócios com os widgets de Gerenciamento de Tarefas Manuais e utilizá-lo para participar nos processos de negócios.
- Usando o Process Portal, os participantes do processo podem se conectar ao servidor Process Center ou a um Process Server em qualquer ambiente de tempo de execução configurado, independentemente de um processo estar sendo desenvolvido, testado ou liberado para um ambiente de produção.
- Gerenciar instâncias de processo do Business Process Execution Language (BPEL) no Business Process Choreographer Explorer ou no Business Space.

Configurações do IBM Business Process Manager

Diferentes configurações do IBM Business Process Manager se correlacionam com pontos de entrada ou estágios típicos no programa de gerenciamento de processo de negócios de uma empresa.

Tabela 1. Configurações do IBM Business Process Manager

Configuração	Fase
Avançada	<p>Transformação</p> <p>Conjunto completo de recursos de gerenciamento do processo de negócios</p> <ul style="list-style-type: none"> • Suporte estendido para automação de processo alto volume • Componentes SOA integrados para a integração de serviço corporativo extensiva, orquestração
Padrão	<p>Programa</p> <p>Configurado para projetos de gerenciamento de processo de negócios típicos</p> <ul style="list-style-type: none"> • Para programas de melhoria multiprojetos, com alto envolvimento de negócios • Suporte de integração do sistema básico • Rápido time-to-value e produtividade do usuário aprimorada
Expresso	<p>Projeto</p> <p>Configurado para o primeiro projeto de gerenciamento de processos de negócios</p> <ul style="list-style-type: none"> • Rápido time-to-value: produtividade do usuário melhorada • Preço de entrada baixo • Fácil instalação e configuração

Recursos de Configuração do IBM Business Process Manager

Entenda os produtos e os recursos oferecidos pela IBM para o gerenciamento de processo de negócios e escolha o correto para sua empresa.

O IBM Business Process Manager é uma plataforma de BPM única que combina recursos centralizados em pessoas e centralizado em integração em um produto unificado. Diferentes configurações do produto estão disponíveis para usuários diferentes e satisfazem necessidades diferentes na empresa. As configurações do produto podem ser combinadas para autoria colaborativa e ambientes de tempo de execução implementados em rede.

Tabela 2. Recursos de Configuração do IBM Business Process Manager

Recurso	Avançado	Padrão	Expresso
Execução compatível com o WebSphere Lombardi Edition	X	X	X
Process Designer (BPMN)	X	X	X
Edição colaborativa / Reprodução imediata	X	X	X
Interfaces com o usuário de "coach do processo" interativas	X	X	X
Regras de negócios baseadas em ODM	X	X	X
Process Portal	X	X	X
Monitoramento e relatório em tempo real	X	X	X
Analíticas & otimizador de desempenho	X	X	X
Performance Data Warehouse	X	X	X
Process Center / repositório de ativo compartilhado	X	X	X
Autores de processo e usuários finais ilimitados	X	X	200 usuários / 3 autores
Alta disponibilidade: armazenamento em cluster e núcleos ilimitados	X	X	<ul style="list-style-type: none">• Produção de 4 núcleos• desenvolvimento de 2 núcleos• Nenhum cluster
Execução compatível com WebSphere Process Server	X		
Integration Designer (BPEL / SOA)	X		
Barramento de serviço corporativo (ESB) integrado	X		
Suporte a transações	X		
Adaptadores de integração	X		
Interface com o usuário do Business Space flexível	X		
Suporte de plataforma avançado (Linux on System z, IBM AIX, Solaris)	X	X	*Nota

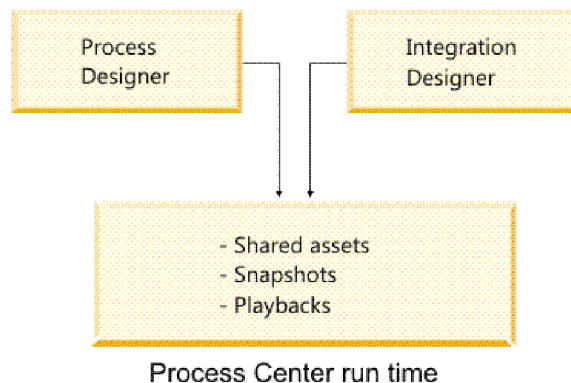
Nota: O IBM BPM Express é suportado no AIX apenas para clientes do IBM Master Data Management (MDM).

O Repositório do Process Center

O Process Center inclui um repositório para todos os processos, serviços e outros ativos criados nos ambientes de criação IBM Business Process Manager, Process Designer e Integration Designer.

O Process Center é um componente de software executado como um servidor no qual o Process Designer e o Integration Designer compartilham ativos, permitindo o desenvolvimento de processos de negócios de maneira cooperativa e altamente interativa.

No diagrama a seguir, você visualiza diversos componentes relacionados que juntos permitem que você construa processos de negócios complexos.



O console do Process Center fornece as ferramentas de que você precisa para manter o repositório.

- A partir do Process Center console, é possível criar aplicativos de processo e kits de ferramentas e conceder a outros usuários acesso a esses aplicativos de processos e kits de ferramentas.
- Nos ambientes de criação, é possível criar modelos de processos, serviços e outros ativos dentro dos aplicativos de processos.
- O Process Center inclui dois servidores, o servidor Process Center e o servidor Performance Data Warehouse. O Servidor Process Center permite que os desenvolvedores que estão trabalhando no Process Designer executem seus aplicativos de processos e armazenem dados de desempenho para teste e reprodução durante os esforços de desenvolvimento. O Performance Data Warehouse recupera dados rastreados do servidor Process Server ou Process Center em intervalos regulares.
- A partir do console Process Center, os administradores instalam aplicativos de processo que estão prontos para testes ou para produção nos servidores de processos naqueles ambientes.
- A partir do console Process Center, os administradores gerenciam as instâncias em execução dos aplicativos de processos nos ambientes configurados.

O console do Process Center fornece um local conveniente no qual criar e manter contêineres de alto nível tais como aplicativos de processo e kits de ferramentas. Os administradores que não trabalham ativamente na visualização Designer podem usar o console do Process Center para fornecer uma estrutura na qual analistas e desenvolvedores de BPM podem construir seus processos e implementações subjacentes. Uma outra tarefa primária para administradores é gerenciar o acesso ao repositório do Process Center configurando a autorização apropriada para usuários e grupos.

Usuários com autorização apropriada podem executar algumas tarefas administrativas diretamente no Process Designer e no Integration Designer. Por exemplo, um desenvolvedor com acesso de gravação ao aplicativo de processo que deseja capturar o estado de todos os recursos do projeto em um estágio significativo de desenvolvimento pode criar uma captura instantânea ao trabalhar na visualização Designer.

Process Server e Ambientes de Tempo de Execução

O Process Server fornece um ambiente de tempo de execução de BPM único que pode suportar uma variedade de processos de negócios, orquestração de serviços e recursos de integração.

Em seus ambientes de autoria, o servidor de processos integrado no Process Center permite executar processos conforme você os constrói. Quando estiver pronto, será possível instalar e executar esses mesmos processos nos servidores de processos em seus ambientes de tempo de execução. O componente Business Performance Data Warehouse coleta e agrega dados do processo de processos em execução nos servidores de processos. É possível usar estes dados para melhorar seus processos de negócios.

O Console Administrativo do Processo permite gerenciar os servidores de processos em seus ambientes de tempo de execução, por exemplo, temporariedade, teste, produção, bem como o servidor de processos que faz parte do Process Center.

Ambientes de Criação

O IBM Business Process Manager Advanced oferece dois ambientes de autoria. Use o IBM Process Designer para modelar de maneira eficiente processos de negócios que envolvem tarefas manuais. Use o IBM Integration Designer para construir serviços que estão autocontidos ou que chamam outros serviços existentes como serviços da Web, aplicativos de recurso corporativo ou aplicativos em execução no CICS e no IMS.

- “Process Designer”
- “Integration Designer” na página 7

Process Designer

O Process Designer está disponível em todas as edições do produto. O IBM Business Process Manager Advanced também oferece ao Integration Designer seus editores e adaptadores associados.

Um processo é a principal unidade de lógica no IBM Business Process Manager. É o contêiner para todos os componentes de uma definição de processo, incluindo serviços, atividades e gateways; timer, mensagem e linhas de sequência, regras e variáveis. Quando você modela um processo, está criando uma Business Process Definition (BPD) reutilizável. Ambos, o Process Designer e o Integration Designer podem criar modelos de processo que podem conter tarefas manuais.

O Process Designer ajuda a desenvolver processos de negócios. Com uma ferramenta orientada a gráficos fácil de usar, é possível criar uma sequência de ações que compõem um processo de negócios e você pode redefinir esse processo no passar do tempo conforme as circunstâncias mudam. Se uma ou mais atividades requerem acesso a grandes sistemas ou serviços de backend que fornecem dados para o processo de negócios, por exemplo para obter informações sobre clientes, é possível atender a essa necessidade usando o Integration Designer. Usando uma interface simples, uma atividade no Process Designer pode chamar um serviço criado no Integration Designer. Esse serviço pode usar fluxos de mediação para transformar, rotear e aprimorar dados e adaptadores para chegar a muitos sistemas de backend da maneira padrão. Em resumo, Process Designer foca no processo de negócios e Integration Designer foca nos serviços automatizados para complementar o processo de negócios. Consulte Introdução ao IBM Process Designer.

Todos os projetos do Process Designer estão contidos nos aplicativos em processo. Você armazena esses aplicativos de processo e artefatos associados no repositório do Process Center. Aplicativos de processo podem compartilhar recursos que foram colocados em kits de ferramentas.

O IBM Business Process Manager fornece várias interfaces com o usuário para permitir que você modele, implemente, simule e inspecione processos de negócios. Você cria e gerencia aplicativos de processo, kits de ferramentas, faixas e capturas instantâneas no Process Center Console. É possível criar modelos de processo, relatórios e serviços simples no Process Designer. É possível executar e depurar processos no Inspetor. E você pode executar simulações no Otimizador.

Os aplicativos de processo desenvolvidos no Process Designer podem ser executados a qualquer momento no servidor Process Center ou salvos em uma captura instantânea e implementados no Process Server. O mesmo é válido para os serviços desenvolvidos no Integration Designer e associados com aplicativos de processo.

Um ponto de extensão do autenticador de login é definido no Process Designer para abrir um ponto de customização para que a lógica de login do lado do cliente trate dos requisitos especiais de autenticação do lado do servidor. Quando a autenticação é acionada, o Process Designer recupera o autenticador e chama a lógica de login. O ponto de extensão do autenticador é fornecido no formato de plug-in do

Eclipse. Para utilizar o plug-in do autenticador, consulte: Instalando o IBM Process Designer.

Integration Designer

O Integration Designer fornece editores e auxílios para ajudar desenvolvedores a criarem processos e serviços automatizados complexos (como módulos SCA, mediações e processos BPEL). Ele está disponível como parte do pacote IBM Business Process Manager Advanced ou como um conjunto de ferramentas independente para outros usos.

O IBM Integration Designer foi projetado como um ambiente de desenvolvimento de integração completo para esses aplicativos integrados de construção. Os aplicativos integrados não são simples. Eles podem chamar aplicativos no EIS (Enterprise Information Systems), envolver processos de negócios por meio de departamentos ou corporações e chamar aplicativos local ou remotamente gravados em uma variedade de idiomas e em execução em diversos sistemas operacionais. Os componentes são criados e montados em outros aplicativos integrados (ou seja, aplicativos criados a partir de um conjunto de componentes) por meio de editores visuais. Os editores visuais apresentam uma camada de abstração entre os componentes e suas implementações. Um desenvolvedor que use as ferramentas pode montar um aplicativo integrado sem conhecimento detalhado da implementação subjacente de cada componente.

As ferramentas do Integration Designer são baseadas em uma arquitetura orientada a serviços. Componentes são serviços e um aplicativo integrado que envolve muitos componentes também é um serviço. Os serviços criados estão em conformidade com os padrões líderes de mercado. Processos BPEL, que também se tornam componentes, são criados de forma semelhantes com ferramentas visuais de fácil utilização que estão em conformidade com o Business Process Execution Language padrão de mercado.

No paradigma do Integration Designer, os componentes são montados em módulos. Importações e exportações são usadas para compartilhar dados entre módulos. Artefatos colocados em uma biblioteca podem ser compartilhados entre módulos.

Módulos e bibliotecas podem ser associados a um aplicativo de processo para uso com o Process Center e podem ser usados como serviços por processos criados no Process Designer. Em tais casos, eles também podem ser implementados com o aplicativo de processo.

Como alternativa, módulos e bibliotecas podem ser implementados diretamente no ambiente de teste ou no Process Server. É possível usar módulos de mediação para criar fluxos de mediação, os quais é possível implementar no WebSphere Enterprise Service Bus ou no Process Server.

O IBM Integration Designer também fornece a capacidade de criar tipos de dados e mapas XML que podem ser implementados no dispositivo WebSphere DataPower. Também é possível transferir arquivos para e a partir do WebSphere DataPower.

Ferramentas de Administração

O IBM Business Process Manager inclui um conjunto de ferramentas de administração para ajudar a realizar tarefas que variam da instalação e gerenciamento de capturas instantâneas à administração de processos e trabalho com os recursos em seu ambiente de TI.

Ferramentas de Linha de Comandos

O IBM Business Process Manager fornece ferramentas da linha de comandos, interfaces de script e interfaces de programação para administrar o ambiente de tempo de execução.

- Ferramentas de linha de comandos são programas simples executados a partir de um prompt de linha de comandos do sistema operacional para execução de tarefas específicas. Utilizando essas ferramentas, você pode iniciar e parar servidores de aplicativos, verificar status de servidores, incluir ou remover nós e outras tarefas.
- O programa de script administrativo do WebSphere (wsadmin) é um ambiente de interpretador de comandos não gráfico que permite que você execute opções administrativas em uma

linguagem de script (Jython ou Jacl) e envie programas de linguagem de script para execução. Ele suporta as mesmas tarefas que o console administrativo, bem como muitas tarefas do console do Process Center. A ferramenta wsadmin foi projetada para ambientes de produção e operações não assistidas.

- Interfaces de programação administrativas são um conjunto de classes e métodos Java sob a especificação Java Management Extensions (JMX) que fornecem suporte para administrar a Service Component Architecture (SCA) e objetos de negócios. Cada interface de programação inclui uma descrição da sua finalidade, um exemplo que demonstra como utilizar a interface ou classe e referências às descrições do método individual.

Console do Process Center

O console do Process Center fornece um local conveniente para que os usuários criem e mantenham itens da biblioteca de alto nível como aplicativos de processos e kits de ferramentas. Ele ajuda a fornecer uma estrutura em que os analistas e os desenvolvedores da BPM podem construir seus processos e implementações subjacentes. Além disso, o console do Process Center fornece ferramentas para manter o repositório, incluindo a configuração da autorização apropriada para usuários e grupos.

Acesse o console do Process Center por meio de um navegador da web (por exemplo, <http://host:9080/ProcessCenter>).

Process Admin Console

O Process Admin Console é usado para administrar os servidores de processo em seu ambiente, incluindo os usuários e capturas instantâneas instalados para cada servidor. Além disso, ele fornece ferramentas para ajudar a gerenciar filas e caches.

O Process Admin Console inclui o Inspetor de Processos, uma ferramenta para visualizar e gerenciar instâncias de processo para aplicativos de processo que estão em execução em um servidor de processo específico.

Acesse o Process Admin Console por meio de um navegador da web (por exemplo, <http://host:9080/ProcessAdmin>).

Console Administrativo de Desempenho dos Negócios

O Console Business Performance Admin Console inclui ferramentas para gerenciar os Performance Data Warehouses em seu ambiente. É possível usar esta ferramenta para gerenciar filas de servidor e desempenho do servidor do monitor.

Acesse o Business Performance Admin Console por meio de um navegador da web (por exemplo, <http://host:9080/PerformanceAdmin>).

Console Administrativo do WebSphere Application Server

O console administrativo é usado para administrar aplicativos, serviços e outros recursos em uma célula, nó, servidor ou escopo de cluster. O console pode ser usado com servidores independentes e com gerenciadores de implementação que gerenciam todos os servidores em uma célula em um ambiente de rede.

Se você instalou um perfil independente, você possui um nó único no seu próprio domínio administrativo, conhecido como uma célula. Use o console administrativo para gerenciar aplicativos, barramentos, servidores e recursos nesse domínio administrativo. Da mesma forma, se você tiver instalado e configurado uma célula de implementação de rede, você tem um nó do gerenciador de implementação e um ou mais nós gerenciados na mesma célula. Use o console administrativo para gerenciar aplicativos, configurar nós gerenciados na célula e monitorar e controlar esses nós e seus recursos.

Acesse esse console por meio de um navegador da web (por exemplo, <http://host:9060/ibm/console> ou <https://host:9043/ibm/console>).

Business Process Choreographer Explorer e Business Process Archive Explorer

Dependendo da função de usuário, é possível usar essas interfaces do cliente para gerenciar processos BPEL e tarefas manuais criadas no IBM Integration Designer, trabalhar com as tarefas

designadas, visualizar processos BPEL e tarefas manuais concluídos que estão em um banco de dados de archive ou excluir processos e tarefas do archive.

Widgets de Administração

Os widgets de Administração oferecem uma maneira de gerenciar e monitorar determinados componentes de sua solução geral do gerenciamento de processo de negócios, incluindo módulos e serviços do Advanced Integration Service. Use esses widgets em um espaço de negócios para fornecer visibilidade para seu aplicativo de serviço e módulos e para responder perguntas como esta:

- Quais serviços são consumidos em um módulo ou expostos por ele e quais são o tempo de resposta e o rendimento em um período de tempo definido para esses serviços?
- Qual é o status de um módulo?
- Há algum evento com falha no módulo?
- As políticas de mediação são associadas ao módulo?
- Quais processos BPEL e tarefas manuais são usados em um módulo?
- Há algum calendário de negócios ou regras de negócios no módulo?

Use um ou mais dos widgets para obter uma captura instantânea do funcionamento geral do sistema de sua solução de negócios, incluindo o status de sua topologia (ambientes de implementação, clusters), aplicativos de sistema (por exemplo, o gerenciador de eventos com falha ou o Business Process Choreographer), origens de dados, mecanismos do sistema de mensagens e filas do sistema de mensagens.

Business Process Rules Manager

O gerenciador de regras de processo de negócios é uma ferramenta na web que ajuda o analista de negócios na procura e modificação de valores de regras de negócios. A ferramenta é uma opção do IBM Process Server que pode ser selecionada para instalar no tempo de criação de perfil ou após a instalação do servidor.

Acessibilidade no IBM Business Process Manager

Os recursos de acessibilidade ajudam usuários com deficiências físicas, como mobilidade restrita ou visão limitada a utilizar produtos de tecnologia da informação com êxito.

A IBM esforça-se para fornecer produtos que sejam acessíveis à maioria das pessoas, independentemente de idade ou capacidade.

Para obter detalhes sobre recursos de acessibilidade deste produto, consulte Recursos de acessibilidade no IBM Business Process Manager.

Disponibilidade de Idiomas Nacionais no IBM Business Process Manager

O IBM Business Process Manager está disponível em vários idiomas. As listas descrevem o nível de suporte para um determinado idioma.

O IBM Business Process Manager suporta os seguintes idiomas: A documentação pode não estar totalmente traduzida.

- Chinês Simplificado
- Chinês Tradicional
- Tcheco
- Inglês dos EUA
- Francês
- Alemão
- Húngaro

- Italiano
- Japonês
- Coreano
- Polonês
- Português do Brasil
- Russo
- Espanhol

O IBM Business Process Manager fornece suporte parcial para os seguintes idiomas: A documentação pode não estar totalmente traduzida.

- Árabe (traduzido para widgets de tarefa manual BPEL, widgets do Business Process Choreographer Explorer, estrutura do Business Space e widgets do Business Space Monitor)
- Dinamarquês (traduzido para widgets do monitor do Business Space e estrutura de Business Space)
- Holandês (traduzido para Process Designer, Process Center, BPD Modeler, Service Modeler, JSEditor, Process Designer, estrutura do Business Space e widgets do Business Space Monitor)
- Finlandês (traduzido para widgets de monitor do Business Space, estrutura do Business Space, BPD Modeler, Service Modeler, JSEditor e Process Designer)
- Grego (traduzido para o Process Designer, Process Center e Business Space)
- Hebraico (traduzido para tarefas manuais BPEL, Business Process Choreographer Explorer, estrutura do Business Space e widgets do Business Space Monitor)
- Norueguês (traduzidos para widgets de monitor do Business Space e estrutura do Business Space)
- Português-Portugal (Process Designer, Process Center, BPD Modeler, Service Modeler e JSEditor)
- Romeno (traduzido para as operações de tempo de execução)
- Eslovaco (traduzido para Business Space, BPD Modeler, Service Modeler, JSEditor e Process Designer)
- Sueco (traduzido para widgets de monitor do Business Space e estrutura do Business Space)
- Turco (traduzido para o Business Space)

Nota: Para o código de idioma Turco, você deve configurar a entrada **case-insensitive-security-cache** no arquivo `60Database.xml` para **false** para permitir que os nomes de usuários e senhas contenham a letra *i*. O arquivo `60Database.xml` está localizado no diretório `install_root\profiles\profile_name\config\cells\cell_name\nodes\node_name\servers\server_name\process-center\config\system\`.

O IBM Business Process Manager fornece suporte para os usuários ao inserirem sequências bidirecionais no ambiente do Process Designer, nos coaches e no Process Portal. Ele fornece API JavaScript para manipulação de teste de linguagem bidirecional.

Os coaches e o Process Portal suportam o uso dos calendários hebraico e árabe.

Visão Geral do Gerenciamento de Processos de Negócios

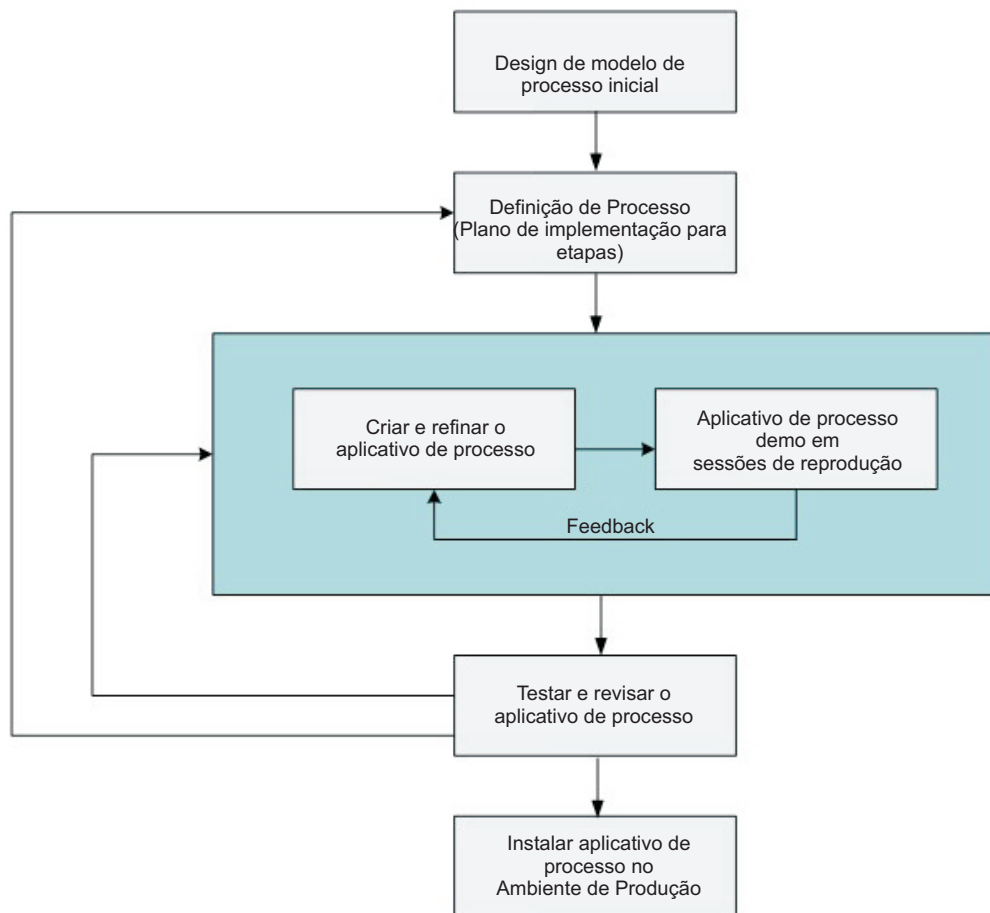
Ao desenvolver processos no Process Designer, é necessário planejar a instalação eventual de seus aplicativos de processo em servidores em seus ambientes de teste e produção.

É possível localizar o Process Designer no IBM Business Process Manager Express, no IBM Business Process Manager Standard e no IBM Business Process Manager Advanced. Nesta seção, nos concentraremos na versão Advanced, que foi projetada para automação de alto volume e uso de serviços complexos desenvolvidos no Integration Designer. Ela possui componentes SOA integrados que podem ser usados para integração de serviço corporativo extenso. A versão Standard pode ser usada por vários profissionais de negócios cooperativamente para desenvolver vários processos sofisticados. Ela possui

integração básica do sistema. A versão Express é para um pequeno número de usuários em um único servidor que estejam sendo introduzidos aos processos de negócios ou que não precisem de acesso a muitos sistemas externos.

O diagrama a seguir mostra o ciclo de vida de um típico esforço de desenvolvimento de processo. Ele inclui etapas para construir e refinar um serviço de instalação para que possa instalar seus aplicativos de processo no ambiente de produção.

Conforme este diagrama mostra, é possível trabalhar exclusivamente em seu ambiente de desenvolvimento. Mas é necessário configurar os Process Servers para seus ambientes de teste e produção.



Visão Geral da Modelagem de Processo

Um processo é a principal unidade de lógica no IBM Business Process Manager. É o contêiner para todos os componentes de uma definição de processo, incluindo serviços, atividades e gateways; timer, mensagem e linhas de sequência, regras e variáveis. Quando você modela um processo, está criando um Business Process Definition (BPD) reutilizável.

Os componentes do processo permitem que você defina o fluxo de trabalho do processo para usuários finais, criando lógica dentro de um processo e integrando com outros aplicativos e origens de dados. Para entender o que ocorre dentro de um processo no tempo de execução, é importante conhecer os componentes que compõem um processo no tempo de design.

Construindo Processos no IBM BPM

Muitos indivíduos diferentes de várias organizações geralmente estão envolvidos nos processos de desenvolvimento usando IBM BPM. A principal preocupação é assegurar que você está compilando a melhor solução possível para reunir os objetivos existente de seu projeto. Para assegurar resultados bem-sucedidos, membros da equipe devem trabalhar juntos para capturar requisitos do processo e desenvolver iterativamente o modelo e suas implementações.

Reutilizando Itens no Process Designer

O Process Designer permite que desenvolvedores de processo reutilizem itens existentes dentro e entre aplicativos de processo. Por exemplo, se você souber que vários serviços já existem, que incluem Coaches e outros itens compartilhados que você e outros desenvolvedores precisam, será possível acessar e reutilizar esses itens incluindo-os em um kit de ferramentas. Em seguida, a partir de seu aplicativo de processo, é possível incluir uma dependência no kit de ferramentas no qual os itens compartilhados residem. Isto permite selecionar um dos serviços existentes ao escolher a implementação para uma atividade. Os itens no toolkit também podem ser usados por outros desenvolvedores que trabalham em diferentes aplicativos de processo.

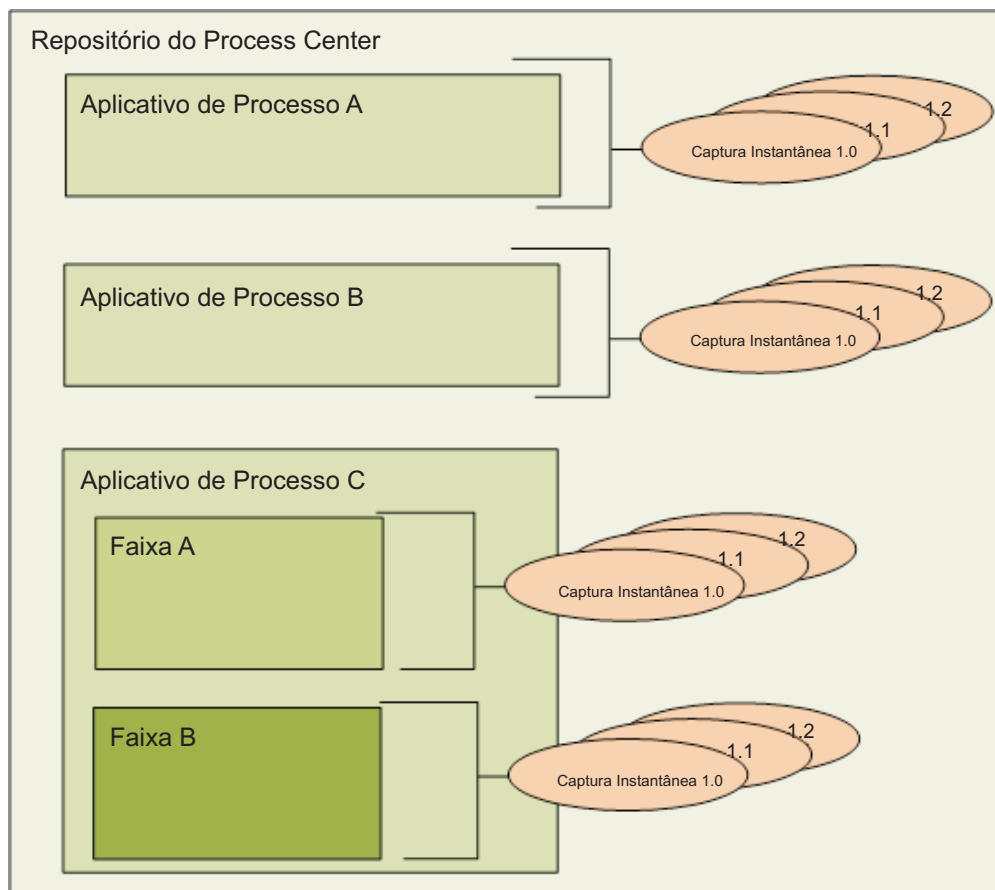
Usando o Designer no IBM Process Designer

A interface do Designer fornece as ferramentas necessárias para modelar seus processos no IBM BPM.

Desenvolvimento de Processo com o Process Center

O IBM Process Center serve como um armazenador central para todos os ativos de projeto criados no Process Designer. Quando diversos clientes do Process Designer se conectam ao Process Center, os usuários podem compartilhar itens, tais como processos e serviços e também podem ver alterações feitas por outros usuários conforme elas ocorrem. O Process Center também pode ser usado como um repositório para ativos criados no IBM Integration Designer.

Quando você está desenvolvendo processos no Process Designer, há uma hierarquia disponível no repositório do Process Center que é projetado para ajudá-lo a gerenciar seus projetos. A figura a seguir fornece uma visão geral conceitual da hierarquia do repositório:



Como você pode ver a partir do diagrama precedente, o repositório do Process Center inclui os seguintes artefatos:

Tipo de conteúdo	Descrição
Aplicativos de Processo	Contêineres para os modelos de processo e implementações de apoio que analistas e desenvolvedores de BPM criam no Designer no IBM Process Designer.
Faixas	Subdivisões opcionais em um aplicativo de processo baseado nas tarefas da equipe ou versões do aplicativo de processo. Quando ativadas, as faixas permitem que ocorra o desenvolvimento paralelo. Os administradores determinam se faixas adicionais são necessárias e, portanto, ativadas para cada aplicativo de processo.
Capturas Instantâneas	Registram o estado dos itens em um aplicativo de processo ou faixa em um momento específico. Geralmente, as capturas instantâneas representam um marco ou são usadas para reproduções ou instalações. É possível comparar capturas instantâneas e reverter em capturas instantâneas anteriores. Se um administrador ativa faixas para um aplicativo de processo, uma captura instantânea é usada como a base para uma nova faixa.

Aplicativos de Processo: Visão Geral

Um aplicativo de processo é um contêiner para modelos de processo e suas implementações de apoio; ele é armazenado no repositório. Após os artefatos terem sido desenvolvidos ou de outra forma criados, eles são montados em um aplicativo de processo.

Os aplicativos de processo contêm alguns ou todos os artefatos a seguir:

- Um ou mais modelos de processo, também chamados de Business Process Definitions (BPDs)
- Referências a kits de ferramentas

- Os serviços requeridos para implementar atividades ou se integrar com outros sistemas, incluindo Serviços de Integração Avançados
- Uma ou mais faixas
- Módulos e bibliotecas do Service Component Architecture (SCA) (desenvolvidos no IBM Integration Designer)
- Qualquer outro item requerido para executar o processo

Para um vídeo introdutório sobre o aplicativo de processo interativo e o desenvolvimento de kit de ferramentas com dicas, capturas instantâneas e faixas, assista “Aplicativo de Processo Interativo e Desenvolvimento de Kit de Ferramentas”, disponível no YouTube ou o Centro de Informações do IBM Education Assistant. Uma transcrição do vídeo está disponível.

Dica, Capturas Instantâneas e Faixas do Aplicativo de Processo

Qualquer mudança feita em um aplicativo de processo é salva dinamicamente no repositório do Process Center na dica, que é a versão de trabalho atual do aplicativo de processo. É possível usar sessões de reprodução na dica para testar e gerenciar instantaneamente a versão de trabalho atual do aplicativo de processo.

O aplicativo de processo permanece no nível de dica até você decidir criar uma captura instantânea, que registra o estado de itens da biblioteca em um aplicativo de processo ou faixa em um momento específico. Geralmente, você obtém uma captura instantânea sempre que está pronto para testar a integração ou que deseja instalar o aplicativo de processo em um servidor do centro de processo ou um servidor de processos para teste, temporariedade ou produção.

Nota: A dica é uma captura instantânea especial; é o único tipo de captura instantânea em que é possível alterar o conteúdo, mas é possível executá-la apenas no servidor do Process Center. Não é possível instalar uma dica em um servidor de processos.

Por padrão, cada aplicativo de processo possui uma única faixa, chamada Principal. Para permitir um desenvolvimento paralelo em um aplicativo de processo, é possível criar faixas adicionais. Essas subdivisões opcionais no aplicativo de processo mantêm as mudanças isoladas. Por exemplo, imagine que sua empresa esteja no processo de refazer a marca; durante essa transição, os aplicativos de processo atuais devem ser mantidos enquanto novas versões estão sendo desenvolvidas com base na identidade corporativa atualizada. Nesta situação, uma equipe pode estar fazendo correções menores na versão atual de um aplicativo de processo (na faixa Principal) enquanto outra equipe está construindo uma nova versão do aplicativo de processo em uma faixa separada.

Kits de Ferramentas para Aplicativos de Processo

Kits de ferramentas são contêineres que armazenam itens de biblioteca (por exemplo, BPDs) para serem reutilizados por aplicativos de processo ou por outros kits de ferramentas. Aplicativos de processo podem compartilhar itens de biblioteca de um ou mais kits e kits podem compartilhar itens de biblioteca de outros kits. Se você tiver acesso a um kit de ferramentas, é possível criar uma dependência dele e usar os itens de biblioteca desse kit de ferramentas no aplicativo de processo.

Aplicativos de Processo e Aplicativos em Nível de Negócios

Um aplicativo de processo possui um aplicativo em nível de negócios (BLA), que age como um contêiner para o aplicativo de processo e seus recursos (recursos incluem coisas como módulos, kits de ferramentas e bibliotecas SCA). Cada captura instantânea de aplicativo de processo possui seu próprio BLA. Muitas das tarefas de administração para uma captura instantânea (por exemplo, parar ou iniciá-la em um servidor de produção) são feitas no nível do BLA, permitindo a administração mais rápida e simples da captura instantânea e de todos os seus recursos.

Aplicativo de Processo Interativo e Desenvolvimento de Kit de Ferramentas

Tabela 3. Apresentação

Cenário	Áudio	Ação na tela
1	Bem-vindo a esta introdução ao aplicativo de processo interativo e desenvolvimento de kit de ferramentas no IBM Business Process Manager.	A tela de boas-vindas mostra o título do vídeo, <i>Aplicativo de processo interativo e desenvolvimento de kit de ferramentas</i> e o subtítulo <i>Saiba sobre o uso das principais capturas instantâneas e faixas</i> . Copyright 2013, IBM Corporation.
2	Neste vídeo, você aprenderá sobre aplicativos de processo e kits de ferramentas e como é possível usar dicas, capturas instantâneas e faixas para gerenciar seu desenvolvimento e implementação.	Uma lista com marcadores dos tópicos cobertos no vídeo é exibida.
3	Os processos evoluem com o tempo conforme as necessidades de sua mudança de organização. Conforme seus aplicativos de processo e kits de ferramentas passam pelo ciclo de desenvolvimento, teste e produção, é possível gerenciar ativos de processo e versões com o IBM Business Process Manager.	Um diagrama do ciclo de design iterativo é mostrado. As setas movem-se pelo diagrama para mostrar a natureza cíclica do processo de design.
4	No IBM Business Process Manager, os aplicativos de processo servem como contêineres para os modelos de processo e implementações de apoio que os desenvolvedores criam no IBM Process Designer.	A guia Aplicativos de Processo no console do Process Center é mostrada. O cursor passa sobre o nome de um aplicativo de processo e sobre um link para onde o usuário pode abri-lo no IBM Process Designer. O texto na tela "Aplicativo de Processo: contêiner instalável para modelo de processo" é mostrada.
5	Os kits de ferramentas são contêineres que retêm os mesmos artefatos que os aplicativos de processo. Entretanto, diferente dos aplicativos de processo, eles não podem ser instalados e executados no servidor de processos. Os kits de ferramentas retêm artefatos que são reutilizáveis por um ou mais aplicativos de processo. Os kits de ferramentas são instalados indiretamente com cada aplicativo de processo que se refere a eles.	A guia Kits de Ferramentas no console do Process Center é mostrada. O cursor passa sobre o nome de um kit de ferramentas e sobre um link para onde o usuário pode abri-lo no IBM Process Designer. O texto na tela "Kit de ferramentas: contêiner para artefatos reutilizáveis" é mostrado.

Tabela 4. Dicas, Capturas Instantâneas e Faixas

Cenário	Áudio	Ação na tela
6	O IBM Business Process Manager suporta uma abordagem iterativa para aplicativo de processo e desenvolvimento de kit de ferramentas com recursos como: <ul style="list-style-type: none"> • A dica, que é a versão de trabalho atual • Capturas instantâneas, que registram o estado de itens da biblioteca a um certo momento • Faixas, que são ramificações opcionais em um aplicativo de processo ou kit de ferramentas que podem ser usados para desenvolver diversas versões simultaneamente 	Uma lista com marcadores dos recursos incluídos no IBM BPM que suportam uma abordagem iterativa para aplicativo de processo e desenvolvimento de kit de ferramentas é mostrada.

Tabela 4. Dicas, Capturas Instantâneas e Faixas (continuação)

Cenário	Áudio	Ação na tela
7	Ao desenvolver um aplicativo de processo ou kit de ferramentas no IBM Process Designer, quaisquer mudanças que fizer serão salvas no repositório do Process Center. Esta versão de trabalho atual do seu aplicativo de processo ou kit de ferramentas é chamada de dica.	Um diagrama do processo no IBM Process Designer é mostrado. Um componente de diagrama é aberto e a palavra <i>data</i> é alterada para a palavra <i>date</i> em um Coach. As mudanças são salvas. O texto na tela “Dica: versão de trabalho atual” é mostrado.
8	É possível usar sessões de reprodução na dica para testar instantaneamente a versão de trabalho atual do aplicativo de processo ou kit de ferramentas. É possível executar a dica somente no servidor do Process Center e não é possível instalar uma dica no servidor de processos.	A página principal do diagrama do processo é aberta e uma sessão de reprodução é iniciada. Isso abre a interface do Inspetor e exibe o Coach alterado em uma janela do navegador. O texto na tela “Dica: reprodução para teste instantâneo” é mostrado.
9	Quando tiver concluído a mudança do seu aplicativo de processo ou kit de ferramentas, será possível criar uma captura instantânea dele. A captura instantânea registra o estado de todos os itens em um aplicativo de processo ou kit de ferramentas em um momento específico. Após criar uma captura instantânea, seus processos e itens relacionados ainda estarão disponíveis para serem editados na dica. No Process Designer, capturas instantâneas disponíveis são mostradas sob o Histórico de Revisão. Embora possa visualizar essas capturas instantâneas, elas não serão editáveis.	A janela do navegador é fechada e uma captura instantânea do aplicativo de processo é criada. A interface do Designer é mostrada e o cursor passa sobre o nome da nova ajuda instantânea sob a seção Histórico de Revisão. O texto na tela “Captura Instantânea: registra o estado de todos os itens” é mostrado.
10	Também é possível comparar capturas instantâneas para ver informações como o horário em que foram criadas e quaisquer processos incluídos nelas.	A visualização de comparação de capturas instantâneas é mostrada. O cursor passa sobre a data em que uma captura instantânea foi criada e o nome de um item no processo que foi alterado. O texto na tela “Comparar mudanças entre capturas instantâneas” é mostrado.
11	Ao criar uma nova captura instantânea de um kit de ferramentas, os aplicativos de processos que se referem a esse kit de ferramentas continuarão usando a antiga captura instantânea. Ao abrir o aplicativo de processo no IBM Process Designer para editá-lo, um aviso notificará que uma nova versão do kit de ferramentas está disponível.	O console do Process Center é aberto. A página Kits de Ferramentas é exibida e um kit de ferramentas é aberto. Uma nova captura instantânea do kit de ferramentas é criada. A página Aplicativos de Processo é aberta e um aplicativo de processo é aberto no Process Designer. O cursor passa sobre o aviso que aparece e seus itens de menu relacionados. O texto na tela “Os aplicativos de processo mantêm referências a uma captura instantânea de kit de ferramentas antigo” é mostrado.
12	Quando seu aplicativo de processo ou kit de ferramentas está pronto para ser implementado, um administrador pode instalar uma ou mais capturas instantâneas deles no servidor de processos usando o console do Process Center. Aplicativos de processos instalados podem se referir somente a capturas instantâneas do kit de ferramentas e não à versão atual de trabalho de um kit de ferramentas.	O console do Process Center é aberto e um aplicativo de processo é aberto. O cursor passa sobre um link em uma captura instantânea que um administrador pode clicar para instalar a captura instantânea. O texto na tela “Instalar capturas instantâneas de aplicativo de processo no servidor de processos” é mostrado.

Tabela 4. Dicas, Capturas Instantâneas e Faixas (continuação)

Cenário	Áudio	Ação na tela
13	Os administradores também podem gerenciar capturas instantâneas a partir do console do Process Center, como criar, arquivar e exportar capturas instantâneas.	O menu suspenso próximo ao nome da captura instantânea é aberto e o cursor passa sobre cada opção no menu. O texto na tela “Gerenciar capturas instantâneas usando o console do Process Center” é mostrado.
14	Por padrão, cada aplicativo de processo ou kit de ferramentas possui uma única faixa, chamada Principal. Os administradores podem ativar a criação de faixas adicionais para permitir o desenvolvimento paralelo. Essas subdivisões opcionais mantêm as mudanças isoladas.	A página Gerenciar do aplicativo de processo é aberta. A opção “Permitir que usuários criem faixas neste aplicativo de processo” é marcada. O texto na tela “Faixas: subdivisões opcionais que mantêm mudanças isoladas” é mostrado.
15	Crie uma nova faixa a partir de uma captura instantânea. Isso copia a captura instantânea selecionada na nova faixa. Outras capturas instantâneas da faixa de origem não são copiadas para a nova faixa.	A página Capturas Instantâneas do aplicativo de processo é aberta. Uma nova faixa chamada Faixa 2 é criada. A nova faixa é aberta e o cursor passa sobre o nome da captura instantânea que foi copiada. O texto na tela “Criar uma faixa a partir de uma captura instantânea” é mostrada.
16	Cada faixa possui uma dica distinta ou versão de trabalho atual. Em um aplicativo de processo ou kit de ferramentas com diversas faixas, é possível copiar ativos de uma captura instantânea em uma faixa na dica de outra faixa. Ao fazer isso, os ativos que foram revisados sobrescrevem os ativos que já estão na faixa de destino e os novos ativos são incluídos.	A nova captura instantânea é aberta e uma página com as configurações do aplicativo de processo e itens é exibida. A definição de processo de negócios é mostrado. Os ativos são copiados na dica da faixa Principal. As páginas onde as listas de ativos novos, atualizados e conflituosos apareceriam são mostradas. O texto na tela “Copiar ativos em outras faixas” é mostrado.
17	A hierarquia das capturas instantâneas do aplicativo de processo e faixas no repositório do faixa é mostrada aqui. Neste diagrama, os aplicativos de processo A e B possuem a faixa única padrão, diversas capturas instantâneas e uma única dica. O Aplicativo de Processos C possui duas faixas, cada uma com diversas capturas instantâneas e uma dica. As capturas instantâneas e dica em cada faixa são independentes umas das outras.	Um diagrama que mostra o relacionamento entre dicas, capturas instantâneas e faixas é mostrado. O diagrama mostra as representações dos aplicativos de processo a seguir: <ul style="list-style-type: none"> • Aplicativo de Processo A, que possui uma faixa, quatro capturas instantâneas e uma dica • Aplicativo de Processo B, que possui uma faixa, três capturas instantâneas e uma dica • Aplicativo de Processo C, que possui duas faixas. Uma faixa possui três capturas instantâneas e uma dica e a outra faixa possui duas capturas instantâneas e uma dica.

Tabela 5. Cenário de Uso de Dicas, Capturas Instantâneas e Faixas

Cenário	Áudio	Ação na tela
18	Agora que sabe o que aplicativos de processo e kits de ferramentas são e como usar dicas, capturas instantâneas e faixas, vamos ver quando usá-los. Acompanharemos Robert, um programador de negócios para Empresa ABC.	Uma fotografia de um homem sorrindo é mostrada. A fotografia é titulada “Robert, programador de negócios, Empresa ABC”.

Tabela 5. Cenário de Uso de Dicas, Capturas Instantâneas e Faixas (continuação)

Cenário	Áudio	Ação na tela
19	<p>Robert está desenvolvendo um processo que modela como esta empresa emprega novos funcionários. Seu processo possui a faixa processo padrão, Principal. Robert faz mudanças no processo na dica. Para assegurar que o processo seja executado tranquilamente, ele começa uma sessão de reprodução, que executa a dica no servidor do Process Center.</p>	<p>Um diagrama que representa um ambiente de desenvolvimento de aplicativo de processo é mostrado.</p> <p>Em uma seção chamada Desenvolvimento, um conjunto dos três retângulos aninhados é mostrado. O retângulo externo representa o repositório do Process Center, o retângulo do meio representa o aplicativo de processo e o retângulo interno representa a faixa Principal.</p> <p>À direita da seção Desenvolvimento está uma seção chamada Teste. Essa seção contém um único retângulo que representa o servidor do Process Center.</p> <p>Abaixo da seção Teste está uma seção chamada Produção. Essa seção contém um único retângulo que representa o servidor de processos.</p> <p>Para mostrar Robert fazendo mudanças na dica, um triângulo que representa a dica é mostrado na faixa Principal. Quando Robert executa uma sessão de reprodução, uma seta mostra a dica sendo copiada no servidor do Process Center.</p>
20	<p>Robert e sua equipe terminaram de desenvolver e testar o aplicativo de processo e ele está pronto para ser implementado no ambiente de produção. Para tal, Robert cria uma captura instantânea do aplicativo de processo. Em seguida, Alice, o administrador, instala a captura instantânea no servidor de processos. A Empresa ABC agora pode usar o processo do Robert para orientar a contratação de novos funcionários.</p>	<p>Para mostrar Robert criando uma captura instantânea, uma elipse representando uma captura instantânea aparece próxima à dica na faixa Principal. Para mostrar Alice instalando a captura instantânea, uma seta mostra a captura instantânea sendo copiada no servidor de processos.</p>
21	<p>Posteriormente, a Empresa ABC anuncia que terá uma nova marca e será conhecida no futuro como a Empresa DEF. Durante esta transição, o processo de contratação que Robert criou deverá ser mantido enquanto uma nova versão será desenvolvida com base na identidade corporativa atualizada. A equipe de manutenção faz correções menores no processo original na faixa Principal enquanto Robert e sua equipe criam uma versão de nova marca do aplicativo de contratação na faixa Nova Marca DEF.</p>	<p>Uma captura instantânea adicional aparece para mostrar que o desenvolvimento adicional foi feito no aplicativo de processo. Um retângulo que representa a faixa Nova Marca DEF é incluído dentro do aplicativo de processo. A faixa Nova Marca DEF inicialmente possui uma captura instantânea e uma dica e outra captura instantânea é incluída para mostrar o desenvolvimento adicional na faixa.</p>
22	<p>Ambas as equipes de manutenção e nova marca podem executar sessões de reprodução separadas no servidor de Process Center. Quando os aplicativos de processo estão prontos para serem implementados, cada faixa pode instalar uma ou mais capturas instantâneas no servidor de processos.</p>	<p>As setas mostram a direção das dicas das faixas que estão sendo executadas no servidor do Process Center ao mesmo tempo. Em seguida, as setas mostram capturas instantâneas de cada faixa que está sendo executada no servidor de processos ao mesmo tempo.</p>

Tabela 6. Conclusão

Cenário	Áudio	Ação na tela
23	Neste vídeo, cobriremos aplicativos de processos e kits de ferramentas e como o IBM Business Process Manager suporta o desenvolvimento de processo interativo com dicas, capturas instantâneas e faixas.	Uma lista com marcadores dos tópicos cobertos no vídeo é mostrada.
24	Para obter mais informações sobre o IBM Business Process Manager e usar dicas, capturas instantâneas e faixas, navegue em outros vídeos do Youtube ou visite esses outros recursos oficiais.	Uma lista com marcadores dos recursos adicionais a seguir é mostrada: <ul style="list-style-type: none"> • Canal do Youtube WebSphereEducation • Documentação do IBM Business Process Manager V8.5 • IBM developerWorks • IBM Education Assistant

Executando e Depurando Processos

Usando o Inspetor, desenvolvedores individuais podem executar processos e serviços no Servidor Process Center ou Process Server de tempo de execução remoto.

O Inspetor no IBM Process Designer é a chave para uma abordagem iterativa para desenvolvimento de processo. Uma equipe de desenvolvimento inteira pode usar o Inspetor para demonstrar o design de processo atual e a implementação em sessões de reprodução. As sessões de reprodução ajudam a capturar informações importantes de diferentes partes interessadas em um processo, como gerência, usuários finais e analistas de negócios. Adotar uma abordagem iterativa para o desenvolvimento de processo garante que seus aplicativos de processo atendam aos objetivos e às necessidades de todas as partes envolvidas.

O Inspetor no IBM Process Designer inclui várias ferramentas que permitem concluir tarefas como as seguintes em cada um de seus ambientes configurados:

Tarefa	Descrição
Gerenciar instâncias de processos	Quando executar um processo, é possível visualizar todas as instâncias executadas anteriormente e as instâncias atualmente em execução nos servidores IBM Business Process Manager em seu ambiente. É possível gerenciar as instâncias em execução parando-as e, em seguida, retomando-as, por exemplo. Também é possível gerenciar instâncias executadas anteriormente filtrando ou excluindo registros específicos.
Percorrer e depurar um processo	Para uma instância selecionada, consulte a etapa em execução atualmente e, em seguida, mova-se no processo, avaliando a execução do processo etapa por etapa. Uma exibição em árvore do processo, combinada com indicadores chamados de tokens no diagrama do processo, faz com que você entenda mais fácil aonde está no processo. Você também tem a vantagem de ver as variáveis usadas em cada etapa e seus valores correspondentes (onde aplicável).

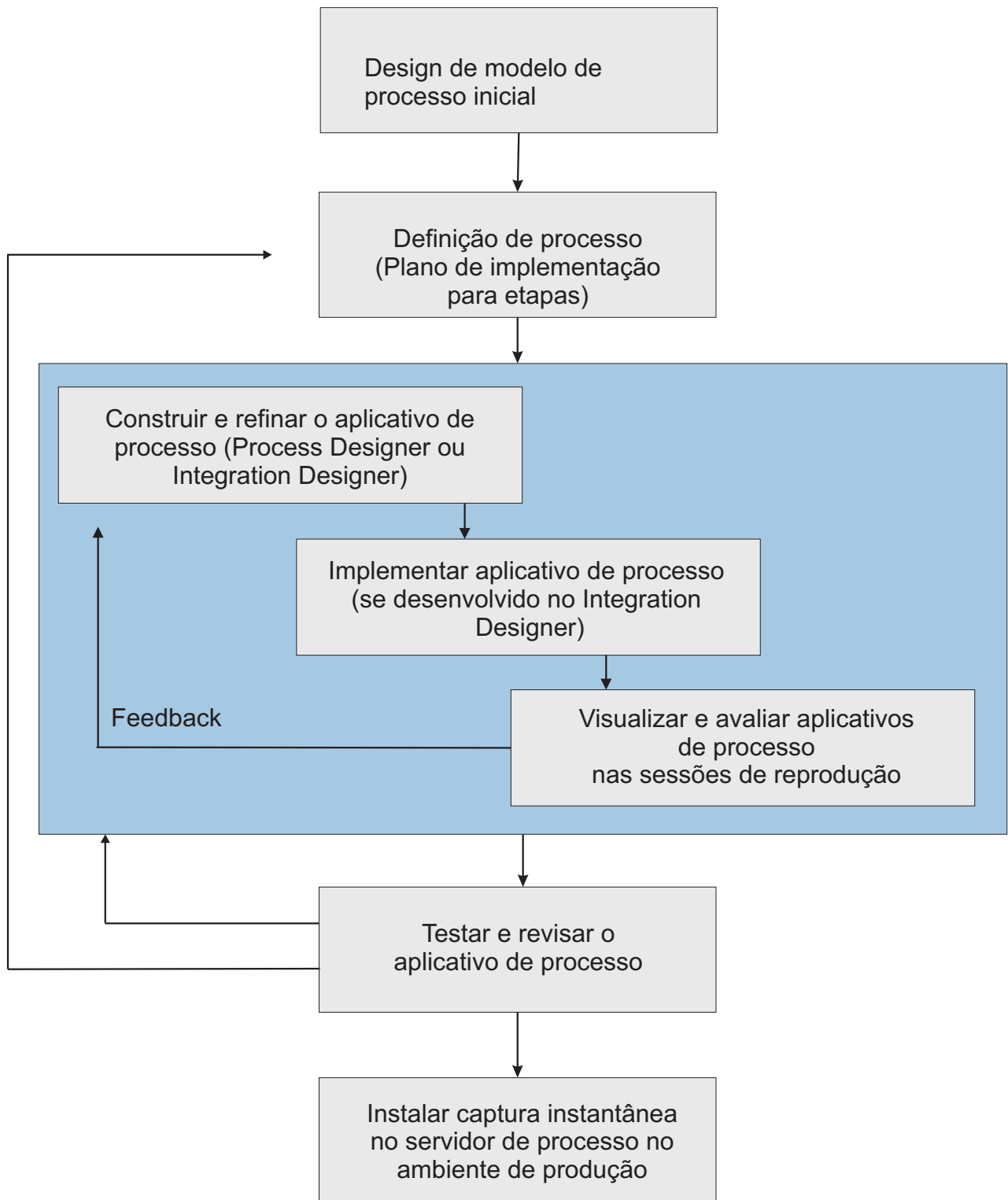
Se estiver trabalhando no IBM Integration Designer, será possível usar o Inspetor se seu projeto estiver associado a um aplicativo de processo. Você também tem outras ferramentas de depuração e teste disponíveis. Para obter informações adicionais sobre essas ferramentas do Integration Designer, consulte "Testando Módulos" e "Usando o Depurador de Integração para Determinação de Problema" nos links relacionados.

Instalando e Gerenciando Aplicativos de Processo

O ciclo de vida do aplicativo de processo inclui a instalação, a administração e a remoção da implementação das capturas instantâneas. As considerações de versão também são parte do ciclo de vida.

Ao desenvolver processos, é possível tirar total vantagem da abordagem iterativa suportada pelas ferramentas dentro do Process Designer. Os processos evoluem com o tempo, inicialmente a partir de um estado de desenvolvimento e, em seguida, até o teste e produção. Mesmo em produção, seus processos podem continuar evoluindo devido às mudanças das necessidades. Estar preparado para o ciclo de vida contínuo de seus processos é importante e ajudará a projetar com eficácia desde o início.

A seguinte figura ilustra uma abordagem iterativa do desenvolvimento de processo.



Uma configuração típica do Business Process Manager inclui três ambientes para suportar o desenvolvimento e a instalação eventual de seus processos.

Ambiente	Descrição
Desenvolvimento	Construa e refine seus aplicativos de processo no IBM Process Designer. Crie seus modelos de processo e implemente as etapas nesses modelos usando o Designer. Usando o Inspector, demonstre seu progresso de desenvolvimento em sessões de reprodução, de forma que você possa avaliar e refinar rapidamente seu protótipo. Usando o console do Process Center, instale seus aplicativos de processo nos servidores de processo de teste ou de produção.
Testar	Usando o console do Process Center, instale seus aplicativos de processo no Process Server em seu ambiente de teste para implementar os testes de controle de qualidade formais. É possível usar o Inspector para ajudar a verificar e resolver problemas.
Produção	Quando todos os problemas relatados a partir de um teste formal forem resolvidos, use o console do Process Center para instalar seus aplicativos de processo no Process Server em seu ambiente de produção. É possível usar o Inspector para investigar e resolver quaisquer problemas relatados no ambiente de produção.

Se você deseja testar, instalar ou administrar uma captura instantânea de aplicativo de processo que tenha conteúdo do IBM Business Process Manager Advanced, o usuário ou o grupo ao qual você pertence deverá ser designado às funções de segurança administrativa Configurator, Operador e Implementador. Se atualmente você não estiver designado para todas essas funções, clique em **Usuários e Grupos** no console administrativo do WebSphere para modificar as funções do usuário e do grupo. Consulte o Funções de Segurança do IBM Business Process Manager.

Estratégias de Liberação e Parcelamento

Para assegurar que os aplicativos de processo implementados e instalados atendam aos padrões de qualidade de sua organização, considere definir uma estratégia de liberação e parcelamento. Quando você tiver identificado os objetivos e requisitos para liberação e parcelamento de aplicativos de processo novos e atualizados, poderá automatizar os processos necessários para aprovar e ativar os programas.

Por exemplo, você pode desejar rotear um processo para vários gerenciadores diferentes em estruturas de relatório diferentes em sua organização. O processo novo ou atualizado poderá ser instalado no seu ambiente de produção e encaminhado para os usuários finais somente depois que cada gerente efetuar sign off do processo. É possível criar e implementar as etapas envolvidas em uma revisão desse tipo no IBM Business Process Manager Advanced para assegurar que todas as diretrizes corporativas tenham sido satisfeitas e que você tenha as assinaturas necessárias. A etapa final da revisão pode ser a equipe de TI notar que o aplicativo de processo aprovado está pronto para instalação.

Criando, Acessando e Incorporando Serviços

Os processos de negócios sempre usam serviços que fornecem as funções necessárias para o processo de negócios. Esses serviços aparecem em um diagrama de processo de negócios como uma atividade ou etapa. Por exemplo, um serviço no Process Designer pode chamar um serviço da web externo ou pode chamar um serviço complexo e automatizado projetado no Integration Designer.

Acessando Serviços Externos a um Aplicativo

Este cenário discute as diferentes formas de acessar os serviços que são externos ao aplicativo e fornece tarefas de alto nível para acessar esses serviços externos.

Nota: Este cenário é aplicável para IBM Business Process Manager Advanced.

Em um aplicativo de negócios integrado, *serviços de negócios* interagem uns com os outros para fornecer uma função necessária. Um serviço de negócios desempenha uma função ou tarefa repetível que contribui para alcançar uma meta de negócio. Mas o trabalho de localizar um serviço e conectar-se a ele não está relacionado à função de negócios. Separar a função de negócios da tarefa de gerenciar conexões de serviço fornece flexibilidade para uma solução.

A interação de serviço começa quando um *solicitante de serviços* envia uma solicitação para o *provedor de serviços* para executar uma função de negócios. Esta solicitação é enviada na forma de uma *mensagem*, que define a função a ser executada. O provedor de serviços executa a função solicitada e envia o resultado em uma mensagem para o solicitante de serviços. Normalmente, as mensagens precisam ser processadas para permitir que os serviços troquem dados e para implementar outras funções de TI de nível inferior que são independentes das funções e dos dados de negócios. Por exemplo, roteamento, conversão de protocolo, transformação, nova tentativa de uma chamada com falha e chamada de serviço dinâmica. Este processamento é conhecido como *mediação*.



Existem dois tipos de módulos no IBM Integration Designer; os módulos (ou módulos de integração de negócios), que são designados principalmente para conter a lógica de negócios (como processos de negócios, regras de negócios e máquinas de estado de negócios) e módulos de mediação, que implementam os fluxos de mediação. Embora haja alguma sobreposição de função entre os dois tipos de módulos, em geral, recomendamos que a lógica de negócios seja isolada em módulos de negócios e que a lógica de mediação seja executada por módulos de mediação.

Mas nem sempre há uma separação clara entre a lógica de negócios e de mediação. Nesses casos, considere a quantidade de *estado* ou dados em variáveis que precisarão ser processados entre as chamadas de serviço. Em geral, se houver pouco ou nenhum processamento de estado necessário, considere o uso de um componente de fluxo de mediação. Se precisar armazenar o estado entre as chamadas de serviço, ou se você tiver dados que precisarão ser armazenados em variáveis e processados, considere usar um componente do processo de negócios. Por exemplo, se estiver chamando diversos serviços e registrando as informações que são retornadas de cada um, para que depois de chamar todos os serviços, você faça algum processamento adicional com os dados retornados, use um processo de

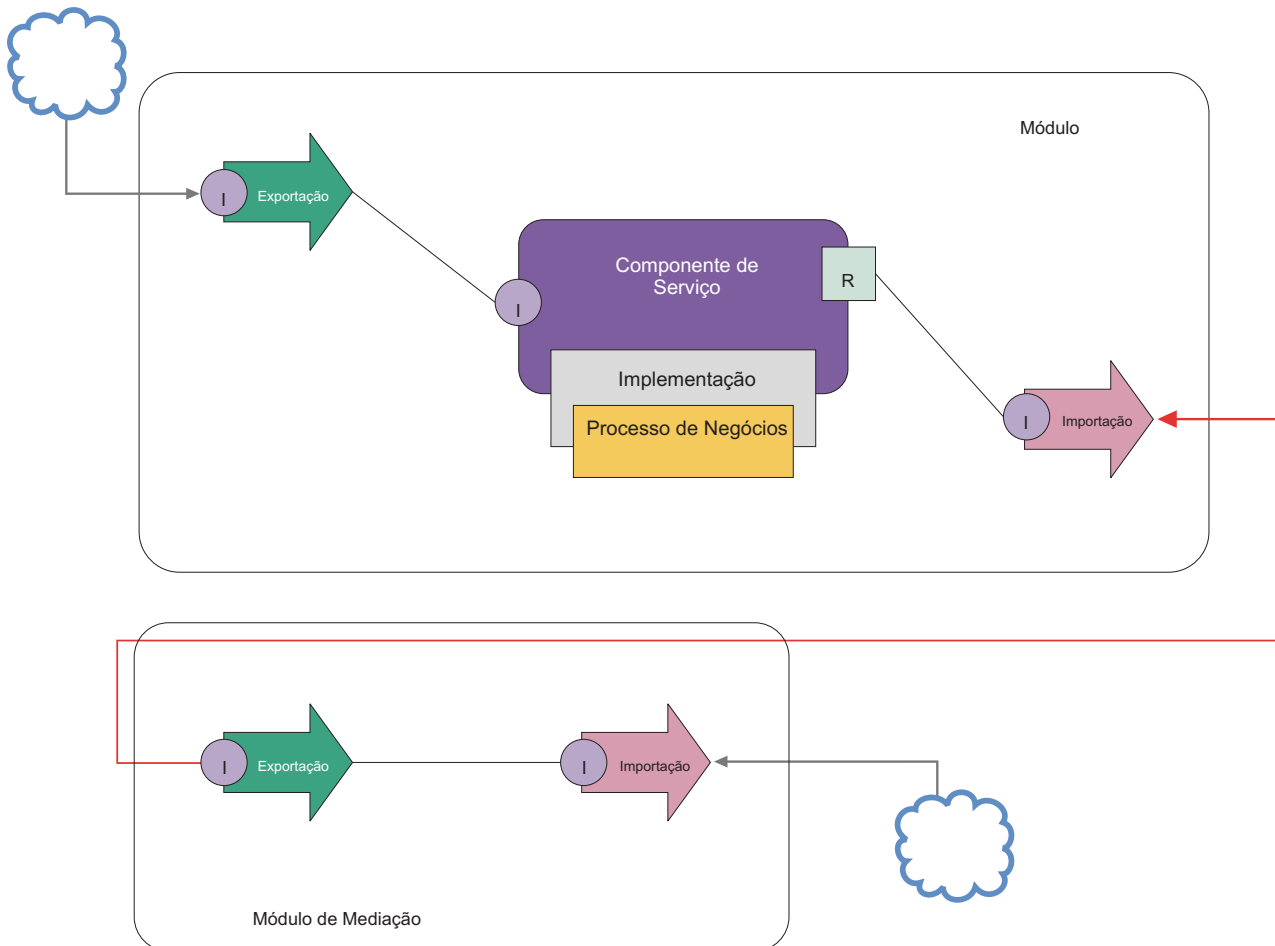
negócios no qual possa designar facilmente as informações retornadas para variáveis. Em outras palavras, quando você tem muito estado, cruzou a linha na lógica de negócios. As seções a seguir ajudam a esclarecer essa orientação.

Não há um cenário de integração e não há resposta tecnicamente errada. As diretrizes que discutimos aqui são boas práticas para permitir flexibilidade e reutilização e são apresentadas para sua consideração. Normalmente, você deve considerar cuidadosamente as vantagens e desvantagens de implementar esses padrões para seus aplicativos de integração de negócios. Vamos considerar algumas situações.

Acessando um Componente SCA

Um exemplo básico de acessar um serviço é quando uma importação chama outro componente SCA, sem exigir nenhuma transformação de dados. Mesmo nesta situação, você poderá acessar o serviço externo a partir de um módulo de mediação, em vez de acessá-lo diretamente a partir de um módulo de negócios. Isto permitiria flexibilidade no futuro, para alterar o terminal em serviço ou as qualidades de serviço ou controle (por exemplo, incluir criação de log) sem impactar os componentes de negócios que consomem o serviço. Esse padrão arquitetural é conhecido como "separação de interesses".

Antes de decidir implementar esse padrão, pese os benefícios do padrão com relação aos efeitos potenciais de sobrecarga introduzida por um outro módulo. Se o seu requisito principal é flexibilidade, e você irá fazer alterações frequentes nos serviços acessados, considere o uso de um módulo separado, conforme mostrado aqui. Se o desempenho for mais importantes, e você estiver disposto a atualizar e reimplementar a lógica de negócios, considere o uso de um único módulo.



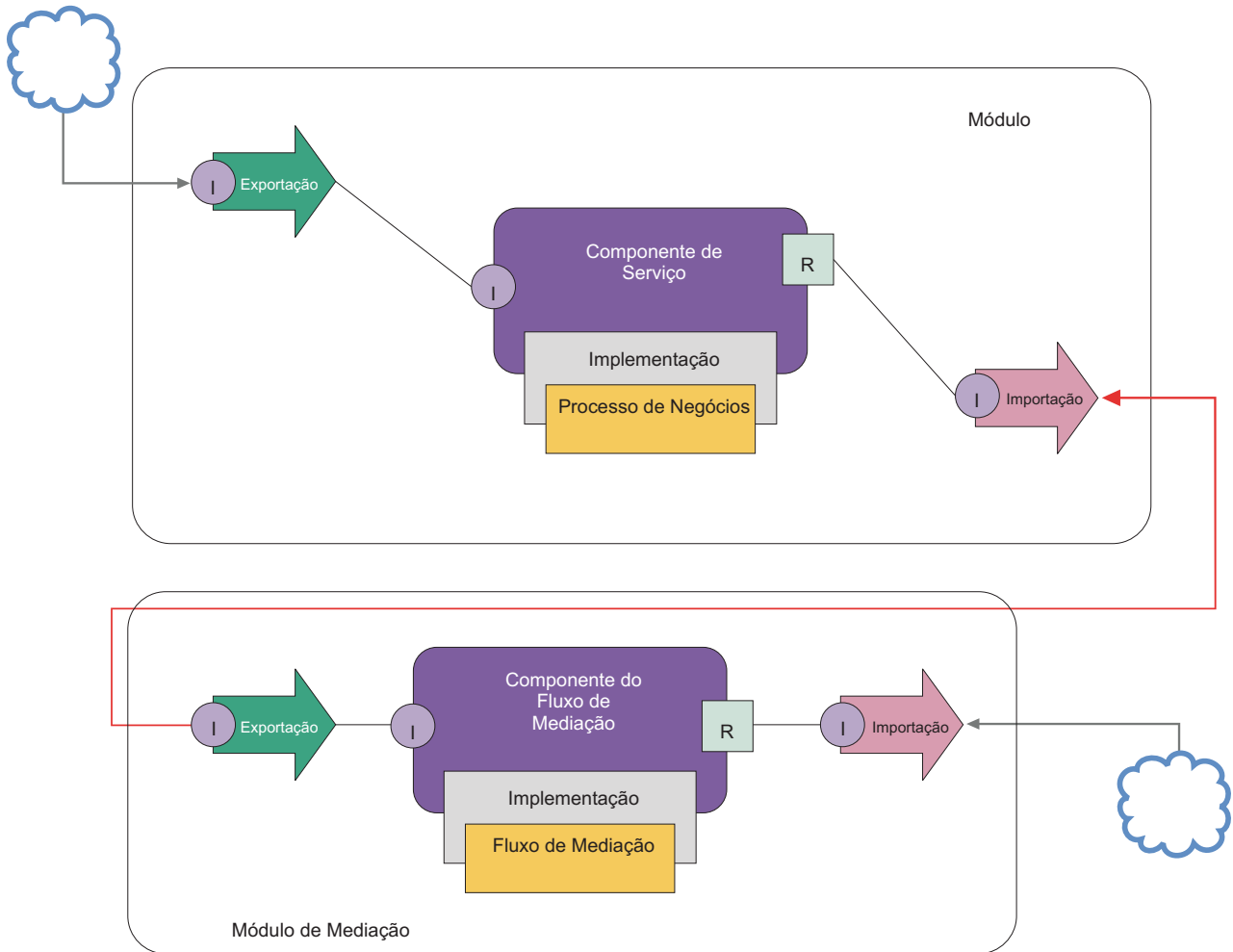
A seguir estão as tarefas de alto nível para alcançar este exemplo.

1. Crie um módulo de mediação. Para obter instruções passo a passo, consulte Criando Módulos de Mediação.
2. No módulo de mediação, crie uma importação com a ligação apropriada para o serviço externo que deseja acessar. Para obter instruções passo a passo, consulte Criando Importações. Para obter informações adicionais sobre ligações, consulte Ligações
3. Crie uma exportação e dê a ela a mesma interface que a importação. Para obter instruções passo a passo, consulte Criando Exportações.
4. Gere uma ligação SCA para a exportação. Para obter instruções passo a passo, consulte Gerando Ligações SCA
5. No conjunto do módulo de mediação, ligue a exportação à importação. Salve o módulo de mediação.
6. Crie um módulo. Para obter instruções passo a passo, consulte Criando um Módulo para Serviços de Negócios
7. Inclua uma exportação e um componente.
8. Na visualização Integração de Negócios, arraste a exportação que você criou no módulo de mediação (na etapa 4) para o conjunto do módulo. Uma importação com a mesma ligação que a exportação será criada.
9. Ligue a exportação ao componente e o componente à importação.
10. Inclua a implementação do componente. Para obter informações sobre tipos de implementação, consulte Implementações

Posteriormente, é possível incluir a lógica de mediação como a criação de log ou o roteamento no módulo de mediação sem afetar o módulo de negócios.

Incluindo Mediação

Às vezes não é suficiente simplesmente chamar um serviço externo. Às vezes é necessário fazer o processamento primeiro, incluindo um módulo de mediação como um intermediário entre o solicitante e o provedor de serviços.



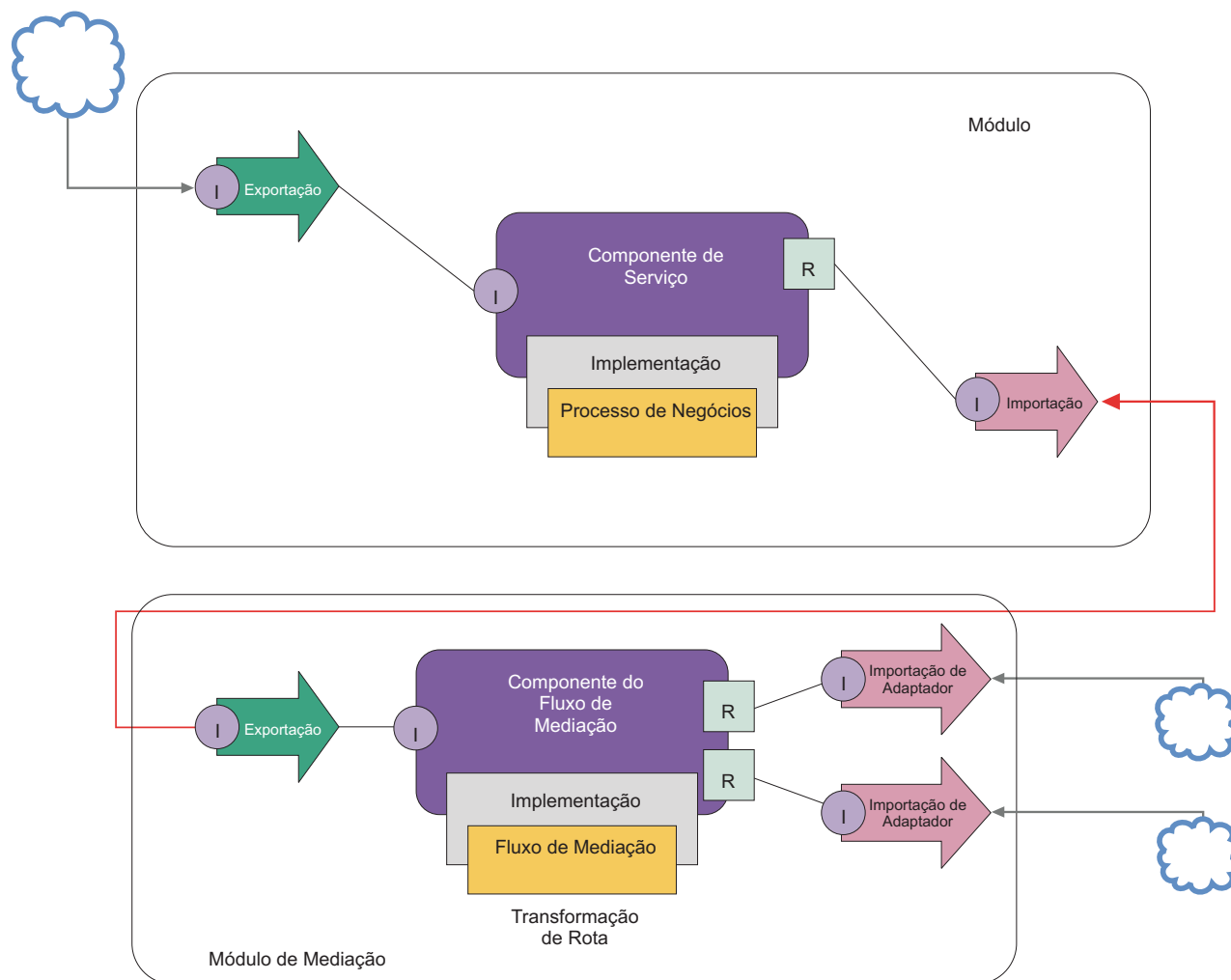
Estas são algumas das funções que o fluxo de mediação intermediário executaria:

- Configurar cabeçalhos de protocolo. Para obter informações adicionais, consulte o tópico Conversão de Protocolo no centro de informações do WebSphere Enterprise Service Bus.
- Transformação da interface ou do parâmetro usando uma primitiva de Mapa de Objeto de Negócios ou de Mapeamento.Transformando Mensagens
- Selecionar um serviço específico a partir de uma lista estática, usando uma primitiva de Filtro de Mensagens.Filtro de Mensagens
- Chamar diversos serviços para agregar os resultados, usando as primitivas Fan Out e Fan In.Agregando e Transmitindo Mensagens
- Lidar com falhas de chamada de serviço tentando novamente o mesmo serviço ou chamando um serviço diferente, usando uma primitiva de Chamada de Serviço. Tentando Novamente uma Chamada de Serviço com Falha
- Roteamento dinâmico escolhendo qual serviço usar no tempo de execução, em vez do tempo de integração, o que permite que os serviços sejam mais fracamente acoplados e as empresas reajam mais rápido à alteração. Novos serviços podem ser incluídos sem tocar nos módulos que foram implementados no ambiente de tempo de execução. O roteamento dinâmico é mais poderoso quando usado com um registro, que requer que a primitiva de mediação de Consulta de Terminal seja usada.Selecionando Terminais Dinamicamente

Acessando os Enterprise Information Systems

Serviços e artefatos em sistemas externos podem ser importados para o Integration Designer. Um assistente descobre aplicativos e dados em um Enterprise Information System (EIS) e permite que você gere serviços a partir dos aplicativos e dados descobertos. Os artefatos gerados são interfaces e objetos de negócios, que podem ser usados por componentes em um módulo.

Usar um módulo de mediação intermediário entre um módulo e um sistema host o torna mais reutilizável. No exemplo abaixo, um fluxo de mediação é usado para rotear para o sistema host correto e para transformar os dados no formato requerido pelo sistema host.



A seguir estão as tarefas de alto nível para este exemplo:

1. Use o assistente de serviço externo para se conectar ao sistema host. O uso do assistente de serviço externo para acessar serviços externos segue um padrão semelhante, independentemente do adaptador que está sendo usado. Para obter informações sobre como usar o assistente de serviço externo, consulte Padrão de Acesso aos Serviços Externos com Adaptadores
2. Crie um módulo. Para obter instruções passo a passo, consulte Criando um Módulo para Serviços de Negócios
3. Inclua uma exportação, um componente e uma importação com ligação SCA. Para obter informações adicionais, consulte Chamando Serviços
4. Inclua uma interface para a exportação e ligue a exportação ao componente.

5. Inclua a implementação do componente. Na implementação, configure uma propriedade que indica qual serviço do host será acessado. Para obter informações sobre tipos de implementação, consulte Implementações
6. Crie um módulo de mediação com uma exportação que possui uma ligação SCA e as mesmas interface que a importação do módulo que você criou na etapa 2.
7. Ligue a exportação a um componente de fluxo de mediação.
8. Crie uma importação para cada sistema host que você deseja acessar, usando o adaptador de saída apropriado na paleta do editor de montagem.
9. Ligue o componente do fluxo de mediação às importações.
10. Implemente o componente do fluxo de mediação. Use uma primitiva Filtro de Mensagem para escolher a importação baseada em um conjunto de propriedades na lógica de negócios e use a primitiva Mapeamento para cada importação de adaptador.
11. No módulo, selecione a exportação do módulo de mediação como o serviço a ser importado no módulo. Para obter informações passo a passo, consulte Chamando um Serviço a Partir de um Outro Módulo.

Posteriormente, é possível fazer uma alteração tal como incluir um adaptador, ou alterar um adaptador para apontar para um sistema host diferente, com impacto mínimo na lógica de negócios.

Acessando Sistemas de Mensagens

Para que seu módulo Service Component Architecture (SCA) se comunique com um cliente de sistema de mensagens JMS, MQ ou JMS do MQ existente, é necessário criar interfaces, objetos de negócios e ligações para importações e exportações. Consulte Mapeando uma Mensagem para uma Interface de SCA.

Os fluxos de mediação usam mensagens, que fornecem acesso às informações de contexto e do cabeçalho além dos objetos de negócios. Se desejar acessar informações do cabeçalho JMS ou uma propriedade JMS customizada, use um fluxo de mediação. Se estiver integrando com um sistema do MQ, e desejar acessar informações do cabeçalho MQ, use um fluxo de mediação.

Criando ou Chamando um Serviço da Web

Os serviços da Web são aplicativos autocontidos que executam funções de negócios, que vão de uma simples consulta até interações de processo de negócios complexas. É possível chamar um serviço da web existente ou desenvolver um novo serviço da web para que se adeque às suas necessidades. Este cenário descreverá as etapas e o apontará para informações adicionais.

Embora você não possa criar todos os seus serviços a partir do início usando o IBM Integration Designer, alguns de seus serviços serão, de fato, criados dessa maneira. Ao trabalhar com o editor de montagem e o editor de processo de negócios para montar os serviços em um processo de negócios, provavelmente você descobrirá que alguns serviços estão ausentes. Pode, portanto, ser útil criar esses serviços ausentes usando as ferramentas do IBM Integration Designer. O inverso também é verdadeiro - após você ter criado um novo processo, poderá decidir que seria útil expor todas operações do processo, ou um subconjunto delas, como um serviço para outras pessoas consumirem.

Nota: Este cenário se aplica a usuários do IBM Integration Designer para IBM Process Server.

Há vários motivos para desenvolver serviços da web usando o IBM Integration Designer:

- A criação de serviços no IBM Integration Designer permite implementar o serviço usando regras de negócios.
- Desenvolver no IBM Integration Designer permite desenvolver um serviço Java™ e expô-lo como um serviço da web e por meio do SCA.
- O mapeamento de interface sem precisar codificar é uma vantagem. É possível retirar todos os mapeamentos de dados do código Java deixando um programa Java de caixa preta simples para o desenvolvedor Java.

- IBM Integration Designer mostra todos os serviços e relacionamentos em um local.
- A capacidade de refatorar também auxiliará no desenvolvimento de serviços da web usando o IBM Integration Designer.

Lembre-se de que os serviços da web não devem ser considerados como a solução para todos os seus problemas de integração. Entretanto, assim como com qualquer outra tecnologia ou abordagem arquitetural, há vantagens inerentes do uso de serviços da web no local correto e no momento certo.

Exportações, Importações e Ligações

O IBM Integration Designer permite importar serviços da web padrão e fazer uso desses serviços em seus aplicativos compostos.

No IBM Integration Designer, use o editor de montagem para desenvolver serviços. Siga o processo padrão para criar módulos, módulos de mediação, bibliotecas e componentes. Em seguida, é possível usar exportações, importações e ligações para compartilhar e acessar esses serviços. As etapas para essas tarefas básicas são listadas abaixo e os links levam a informações mais detalhadas para cada tarefa.

É possível usar uma das duas ligações para serviços da web - uma ligação de serviço da web ou uma ligação HTTP. Uma ligação de serviço da web fornece uma especificação para transmitir mensagens para e de um serviço da web. As ferramentas ajudam a gerar uma ligação de serviço da web automaticamente. Uma ligação HTTP é um protocolo de solicitação e resposta padrão entre clientes e o servidor, conforme definido pelo protocolo HTTP publicado pelo World Wide Web consortium (W3C). Você precisará fornecer algumas informações de configuração de ligação iniciais se usar uma ligação HTTP.

1. Crie uma exportação para publicar serviço do módulo para uso por outros módulos.
2. Gere uma ligação para a exportação.
 - Gere uma ligação de serviço da web para a exportação.
 - Gere uma ligação de exportação HTTP.
3. Crie uma importação para chamar um serviço existente que não faz parte do módulo que você está montando.
 - Gere uma ligação de serviço da web para a importação.
 - Gerando uma ligação de importação HTTP.

Leia o tópico vinculado se desejar Chamar um Serviço da Web a partir de Páginas JavaServer.

Recursos de Desenvolvimento de Serviços da Web

Ao abrir um editor associado ao processo de criação de serviços da web, talvez visualize a janela Confirmar Ativação, que exibe as informações a seguir:

Esta ação requer a
ativação de "Implementação de Serviços da Web".
Ativar a capacidade necessária?

O IBM Integration Designer fornece uma função de filtragem conhecida como *recursos*. Nas configurações de Preferências, funções e ferramentas são categorizadas em recursos e você pode ativar e desativar categorias de recursos ou as funções do subconjunto de qualquer categoria. Consulte Recursos para obter informações adicionais.

Aprenda Mais sobre os Conceitos Chave

Use esta seção como um ponto de início para investigar as tecnologias usadas no IBM Business Process Manager e por ele.

Cenários de Autoria

Use cenários para entender e trabalhar com componentes e produtos da família de gerenciamento de processos de negócios.

Criação de Versões

O ciclo de vida de um aplicativo de processo começa com a criação do aplicativo de processo e continua por um ciclo de atualização, implementação, coimplementação, remoção de implementação e arquivamento do aplicativo de processo. *Versão* é um mecanismo usado para gerenciar o ciclo de vida do aplicativo de processo identificando exclusivamente as versões individuais do aplicativo de processo.

A maneira como essa versão funciona no IBM Business Process Manager depende do que você está implementando – um aplicativo de processo, implementado a partir do repositório no IBM Process Center ou um aplicativo corporativo implementado diretamente a partir do IBM Integration Designer.

Os aplicativos de processo e kits de ferramentas que você implementa em um ambiente de tempo de execução a partir do Process Center são, por padrão, com versão. Para aplicativos corporativos, você pode optar por criar a versão de módulos e bibliotecas no IBM Integration Designer.

Além disso, é possível criar versões de uma tarefa manual ou máquina de estado, para que diversas versões da tarefa ou máquina de estado possam coexistir no ambiente de tempo de execução.

Versão dos Aplicativos de Processo

A versão fornece a capacidade para que o ambiente de tempo de execução identifique capturas instantâneas no ciclo de vida de um aplicativo de processo e possa executar diversas capturas instantâneas simultaneamente em um servidor de processos.

Para entender como os aplicativos de processo são transformados em versão, é importante lembrar-se que um aplicativo de processo é um contêiner que contém vários artefatos usados no aplicativo de processo ou por ele (por exemplo, modelos de processo ou BPDs, referências ao kit de ferramentas, serviços ou rastreios). Qualquer versão é feita nesse nível de contêiner, não no nível dos artefatos individuais. Para aplicativos de processo, isso significa que a versão acontece quando uma captura instantânea é obtida.

É possível comparar capturas instantâneas para determinar diferenças entre as versões. Por exemplo, se um desenvolvedor corrigiu um problema em um serviço e obteve uma captura instantânea de seu aplicativo de processo ou de kit de ferramentas contido nesse ponto e, em seguida, um desenvolvedor diferente fez várias mudanças adicionais no mesmo serviço e obteve uma nova captura instantânea, o gerente de projeto poderá comparar as duas capturas instantâneas para determinar quais mudanças foram feitas quando e por quem. Se o gerente de projeto decidir que as mudanças adicionais no serviço não foram vantajosas, ele poderá reverter para a captura instantânea da correção original.

É possível executar diferentes versões (capturas instantâneas) de um aplicativo de processo simultaneamente em um servidor; durante a instalação de uma nova captura instantânea, remova a original ou deixe-a em execução.

Contexto da Versão

Cada captura instantânea possui metadados exclusivos para identificar a versão (referida como contexto da versão). Você designa o identificador, mas a IBM recomenda usar um sistema de versão numérica de três dígitos no formato <major>.<minor>.<service>. Consulte os tópicos sobre convenções de nomenclatura para obter uma descrição mais detalhada deste esquema de versão.

O IBM Business Process Manager designa um namespace global para cada aplicativo de processo. O namespace global é especificamente a dica do aplicativo de processo ou uma captura instantânea de aplicativo de processo específica. O nome da versão usado pelo servidor não pode ter mais de sete caracteres, portanto, o nome designado será um acrônimo que usa caracteres do nome de captura

instantânea designado. Os acrônimos de captura instantânea serão idênticos aos seus nomes de captura instantânea se os nomes de captura instantânea estiverem em conformidade com o estilo IBM VRM recomendado e não tiverem mais de sete caracteres. Por exemplo, um nome de captura instantânea de 1.0.0 terá um acrônimo de 1.0.0 e um nome de captura instantânea de 10.3.0 terá o acrônimo de 10.3.0. O acrônimo de captura instantânea terá garantia de ser exclusivo no contexto do aplicativo de processo dentro do escopo do servidor Process Center. Por essa razão, não é possível editar o acrônimo da captura instantânea.

Considerações sobre Versão para Aplicativos de Processo em Diversos Clusters

É possível instalar a mesma versão de um aplicativo de processo em diversos clusters dentro da mesma célula. Para diferenciar entre essas diversas instalações da mesma versão do aplicativo de processo, crie uma captura instantânea para cada instalação e inclua um ID de célula exclusivo no nome da captura instantânea (por exemplo, v1.0_cell1_1 e v1.0_cell1_2). Cada captura instantânea é uma nova versão do aplicativo de processo (de uma perspectiva pura de gerenciamento de ciclo de vida), mas o conteúdo e a função são os mesmos.

Quando você instala um aplicativo de processo em um cluster, uma sincronização automática dos nós é executada.

Considerações sobre Versão para Kits de Ferramentas do Process Designer

Lembre-se de que capturas instantâneas de aplicativo de processo são geralmente obtidas quando se está pronto para testar ou instalar. No entanto, capturas instantâneas de kits de ferramentas são geralmente obtidas quando se está pronto para que esse kit de ferramentas seja usado pelos aplicativos de processo. Posteriormente, se desejar atualizar o kit de ferramentas, você deverá obter uma outra captura instantânea de "dica" quando estiver pronto e, então, os proprietários de aplicativos de processo e kits de ferramentas poderão decidir se desejam mover para cima para a nova captura instantânea.

Criando Versão de Módulos e Bibliotecas

Se um módulo ou biblioteca estiver em um aplicativo de processo ou kit de ferramentas, ele(a) assumirá o ciclo de vida do aplicativo de processo ou kit de ferramentas (versões, capturas instantâneas, rastreios e assim por diante). Nomes de módulo e de biblioteca devem ser exclusivos dentro do escopo de um aplicativo de processo ou kit de ferramentas.

Este tópico descreve a versão de módulos e bibliotecas que são usados com aplicativos de processo. Observe, no entanto, que, se você implementar módulos diretamente a partir do IBM Integration Designer para o Process Server, será possível continuar seguindo o procedimento de designação de números da versão para módulos durante a implementação, conforme descrito em "Criando Módulos e Bibliotecas Com Versão".

Um módulo ou uma biblioteca que está associada ao IBM Process Center deve ter suas bibliotecas dependentes no mesmo aplicativo de processo ou em um kit de ferramentas dependente.

A tabela a seguir lista as seleções que podem ser feitas no editor de dependência no IBM Integration Designer quando uma biblioteca está associada a um aplicativo de processo ou kit de ferramentas:

Tabela 7. Dependências para Módulo, Aplicativo de Processo ou Kit de Ferramentas e Bibliotecas Globais

Escopo da Biblioteca	Descrição	Pode depender de . . .
Módulos	Uma cópia desta biblioteca existe no servidor para cada módulo que ele usa.	Uma biblioteca com escopo definido no módulo pode depender de todos os tipos de bibliotecas.

Tabela 7. Dependências para Módulo, Aplicativo de Processo ou Kit de Ferramentas e Bibliotecas Globais (continuação)

Escopo da Biblioteca	Descrição	Pode depender de . . .
Aplicativo de Processo ou Kit de Ferramentas	A biblioteca é compartilhada entre todos os módulos no escopo do aplicativo de processo ou kit de ferramentas. Esta configuração entra em vigor se a implementação é feita através do IBM Process Center. Se a implementação ocorre fora do IBM Process Center, a biblioteca é copiada em cada módulo. Nota: As bibliotecas criadas no IBM Integration Designer versão 8 possuem um nível de compartilhamento de Aplicativo de Processo ou Kit de Ferramentas por padrão.	Uma biblioteca deste tipo pode depender somente de bibliotecas globais.
Global	A biblioteca é compartilhada entre todos os módulos que estão em execução.	Uma biblioteca global pode depender apenas de outras bibliotecas globais. Nota: Você deve configurar uma biblioteca compartilhada do WebSphere a fim de implementar a biblioteca global. Consulte “Dependências de Módulo e Biblioteca” para obter informações adicionais.

Módulos e Bibliotecas Associados com Aplicativos de Processo ou Kits de Ferramentas

Não é necessário criar versão de módulos e bibliotecas associados com aplicativos de processo ou kits de ferramentas.

Os módulos e as bibliotecas que são associados a um aplicativo de processo ou kit de ferramentas não precisam ter versão. Na verdade, não é possível criar uma versão de um módulo ou de uma biblioteca associada a um aplicativo de processo ou kit de ferramentas no editor de dependência. Módulos e bibliotecas associados a um aplicativo de processo ou kit de ferramentas usam capturas instantâneas, uma função no Process Center, para obter o mesmo resultado que uma versão.

As bibliotecas associadas a um aplicativo de processo ou kit de ferramentas não terão um número de versão requerido na seção Bibliotecas do editor de dependência porque nenhuma versão é necessária.

Convenções de Nomenclatura

Uma convenção de nomenclatura é usada para diferenciar as várias versões de um aplicativo de processo conforme ele se move pelo ciclo de vida de atualização, implementação, coimplementação, remoção de implementação e arquivamento.

Esta seção fornece as convenções que são usadas para identificar exclusivamente as versões de um aplicativo de processo.

Um *contexto de versão* é uma combinação de acrônimos que descrevem exclusivamente um aplicativo de processo ou kit de ferramentas. Cada tipo de acrônimo possui uma convenção de nomenclatura. O acrônimo é limitado a um comprimento máximo de sete caracteres do conjunto de caracteres [A-Z0-9_], exceto para o acrônimo de captura instantânea, que também pode incluir um ponto.

- O acrônimo do aplicativo de processo é criado quando o aplicativo de processo é criado. Ele pode ter um máximo de sete caracteres de comprimento.
- O acrônimo de captura instantânea é criado automaticamente quando a captura instantânea é criada. Ele pode ter um máximo de sete caracteres de comprimento.

Se o nome da captura instantânea atender os critérios para um acrônimo de captura instantânea válido, o nome da captura instantânea e o acrônimo serão iguais.

Nota: Ao usar a função de roteamento reconhecida pela versão do componente do fluxo de mediação, nomeie sua captura instantânea para que ela esteja em conformidade com o esquema *<version>.<release>.<modification>* (por exemplo, **1.0.0**). Como o acrônimo de captura instantânea é limitado a sete caracteres, os valores de dígitos são limitados a um máximo de cinco dígitos no total (cinco dígitos mais dois pontos). Portanto, tome cuidado quando os campos de dígitos forem incrementados, porque qualquer coisa além dos primeiros sete caracteres será truncada.

Por exemplo, um nome de captura instantânea **11.22.33** resulta em um acrônimo de captura instantânea **11.22.3**.

- O acrônimo de faixa é gerado automaticamente a partir do primeiro caractere de cada palavra do nome da faixa. Por exemplo, uma nova faixa criada com o nome **My New Track** resultaria em um valor de acrônimo igual a **MNT**.

O nome da faixa padrão e o acrônimo são **Main**. A implementação em um servidor IBM Process Center inclui o acrônimo de faixa no contexto de versão se o acrônimo de faixa não é **Main**.

Uma definição de processo de negócios em um aplicativo de processo é geralmente identificada pelo acrônimo do nome do aplicativo de processo, o acrônimo da captura instantânea e o nome da definição de processo de negócios. Escolha nomes exclusivos para as suas definições de processo de negócios sempre que possível. Quando existirem nomes duplicados, você poderá encontrar os seguintes problemas:

- Você poderá não conseguir expor as definições de processo de negócios como serviços da Web sem alguma forma de mediação.
- Você poderá não conseguir chamar uma definição de processo de negócios criada no IBM Process Designer a partir de um processo BPEL criado no IBM Integration Designer.

O contexto da versão varia, dependendo de como o aplicativo de processo é implementado.

Convenções de Nomenclatura para Implementações do Servidor do Process Center:

No servidor IBM Process Center, é possível implementar uma captura instantânea de um aplicativo de processo bem como uma captura instantânea de um kit de ferramentas. Além disso, é possível implementar a dica de um aplicativo de processo ou a dica de um kit de ferramentas. (Uma *dica* é a versão de trabalho atual de seu aplicativo de processo ou kit de ferramentas.) O contexto de versão varia, dependendo do tipo de implementação.

Para os aplicativos de processo, a dica de aplicativo de processo ou a captura instantânea do aplicativo de processo específica é usada para identificar exclusivamente a versão.

Os kits de ferramentas podem ser implementados com um ou mais aplicativos de processo, mas o ciclo de vida de cada kit de ferramentas é ligado ao ciclo de vida do aplicativo de processo. Cada aplicativo de processo possui sua própria cópia do kit de ferramentas dependente ou kits de ferramentas implementados no servidor. Um kit de ferramentas implementado não é compartilhado entre aplicativos de processo.

Se a faixa associada ao aplicativo de processo for denominada como algo diferente do padrão de **Main**, o acrônimo da faixa também fará parte do contexto da versão.

Para obter informações adicionais, consulte a seção “Exemplos” na página 34, posteriormente neste tópico.

Capturas Instantâneas do Aplicativo de Processo

Para implementações de captura instantânea do aplicativo de processo, o contexto da versão é uma combinação dos seguintes itens:

- Acrônimo do nome do aplicativo de processo
- Acrônimo da faixa do aplicativo de processo (se uma faixa diferente de **Main** for usada)

- Acrônimo da captura instantânea do aplicativo de processo

Kits de Ferramentas Independentes

Para implementações de captura instantânea do kit de ferramentas, o contexto da versão é uma combinação dos itens a seguir:

- Acrônimo do nome do kit de ferramentas
- Acrônimo da faixa do kit de ferramentas (se uma faixa diferente de **Main** for usada)
- Acrônimo da Captura Instantânea do Kit de Ferramentas

Dicas

Dicas do aplicativo de processo são usadas durante o teste iterativo no Process Designer. Elas podem ser implementadas apenas nos servidores Process Center.

Para implementações da dica do aplicativo de processo, o contexto da versão é uma combinação dos itens a seguir:

- Acrônimo do nome do aplicativo de processo
- Acrônimo da faixa do aplicativo de processo (se uma faixa diferente de **Main** for usada)
- "Dica"

As dicas do kit de ferramentas também são usadas durante o teste iterativo no Process Designer. Elas não são implementadas em um servidor de produção.

Para implementações de dica do kit de ferramentas, o contexto da versão é uma combinação dos itens a seguir:

- Acrônimo do nome do kit de ferramentas
- Acrônimo da faixa do kit de ferramentas (se uma faixa diferente de **Main** for usada)
- "Dica"

Exemplos

Os recursos devem ser nomeados e identificados exclusivamente externamente usando o contexto da versão.

- A tabela a seguir mostra um exemplo de nomes que são identificados exclusivamente. Neste exemplo, uma dica de aplicativo de processo usa o nome de faixa padrão (**Main**):

Tabela 8. Dica do Aplicativo de Processo com o Nome de Faixa Padrão

Tipo de nome	Exemplo
Nome do Aplicativo de Processo	Aplicativo de Processo 1
Acrônimo do nome do aplicativo de processo	PA1
Faixa do aplicativo de processo	Principal
Acrônimo da faixa do aplicativo de processo	"" (quando a faixa é Main)
Captura instantânea do aplicativo de processo	
Acrônimo da captura instantânea do aplicativo de processo	Dica

Quaisquer módulos SCA associados a esta dica de aplicativo de processo incluem o contexto de versão, conforme mostrado na tabela a seguir:

Tabela 9. Módulos SCA e Arquivos EAR Reconhecidos pela Versão

Nome do módulo SCA	Nome Reconhecido pela Versão	Nome de EAR/Aplicativo Reconhecido pela Versão
M1	PA1-Tip-M1	PA1-Tip-M1.ear
M2	PA1-Tip-M2	PA1-Tip-M2.ear

- A tabela a seguir mostra um exemplo de uma dica de aplicativo de processo que usa um nome de faixa não padrão:

Tabela 10. Dica de Aplicativo de Processo com Nome de Faixa Não Padrão

Tipo de nome	Exemplo
Nome do Aplicativo de Processo	Aplicativo de Processo 1
Acrônimo do nome do aplicativo de processo	PA1
Faixa do aplicativo de processo	Track1
Acrônimo da faixa do aplicativo de processo	T1
Captura instantânea do aplicativo de processo	
Acrônimo da captura instantânea do aplicativo de processo	Dica

Quaisquer módulos SCA associados a esta dica de aplicativo de processo incluem o contexto de versão, conforme mostrado na tabela a seguir:

Tabela 11. Módulos SCA e Arquivos EAR Reconhecidos pela Versão

Nome do módulo SCA	Nome Reconhecido pela Versão	Nome de EAR/Aplicativo Reconhecido pela Versão
M1	PA1-T1-Tip-M1	PA1-T1-Tip-M1.ear
M2	PA1-T1-Tip-M2	PA1-T1-Tip-M2.ear

Convenções de nomenclatura semelhantes se aplicam às implementações avançadas de captura instantânea e de dica do Kit de Ferramentas. Elas também se aplicam às capturas instantâneas instaladas no Servidor de Processo.

- A tabela a seguir mostra um exemplo de nomes que são identificados exclusivamente. Nesse exemplo, uma captura instantânea do aplicativo de processo usa o nome de rastreo padrão (**Main**):

Tabela 12. Captura Instantânea do Aplicativo de Processo com Nome de Rastreo Padrão

Tipo de nome	Exemplo
Nome do Aplicativo de Processo	Aplicativo de Processo 1
Acrônimo do nome do aplicativo de processo	PA1
Faixa do aplicativo de processo	Principal
Acrônimo da faixa do aplicativo de processo	"" (quando a faixa é Main)
Captura instantânea do aplicativo de processo	Captura Instantânea do Processo V1
Acrônimo da captura instantânea do aplicativo de processo	PSV1

Quaisquer módulos SCA associados a essa captura instantânea de aplicativo do processo incluem o contexto de versão, conforme mostrado na tabela a seguir:

Tabela 13. Módulos SCA e Arquivos EAR Reconhecidos pela Versão

Nome do módulo SCA	Nome Reconhecido pela Versão	Nome de EAR/Aplicativo Reconhecido pela Versão
M1	PA1-PSV1-M1	PA1-PSV1-M1.ear
M2	PA1-PSV1-M2	PA1-PSV1-M2.ear

- A tabela a seguir mostra um exemplo de uma captura instantânea de aplicativo de processo que usa um nome de faixa não padrão:

Tabela 14. Captura instantânea do aplicativo de processo com nome de rastreamento não padrão

Tipo de nome	Exemplo
Nome do Aplicativo de Processo	Aplicativo de Processo 1
Acrônimo do nome do aplicativo de processo	PA1
Faixa do aplicativo de processo	Track1
Acrônimo da faixa do aplicativo de processo	T1
Captura instantânea do aplicativo de processo	Captura Instantânea do Processo V1
Acrônimo da captura instantânea do aplicativo de processo	PSV1

Quaisquer módulos SCA associados a essa captura instantânea de aplicativo do processo incluem o contexto de versão, conforme mostrado na tabela a seguir:

Tabela 15. Módulos SCA e Arquivos EAR Reconhecidos pela Versão

Nome do módulo SCA	Nome Reconhecido pela Versão	Nome de EAR/Aplicativo Reconhecido pela Versão
M1	PA1-T1-PSV1-M1	PA1-T1-PSV1-M1.ear
M2	PA1-T1-PSV1-M2	PA1-T1-PSV1-M2.ear

Convenções de Nomenclatura para Implementações do Process Server:

No Process Server, é possível implementar a captura instantânea de um aplicativo de processo. O acrônimo da captura instantânea do aplicativo de processo é usado para identificar exclusivamente a versão.

Para implementações de captura instantânea do aplicativo de processo, o contexto da versão é uma combinação dos seguintes itens:

- Acrônimo do nome do aplicativo de processo
- Acrônimo da captura instantânea do aplicativo de processo

Os recursos devem ser nomeados e identificados exclusivamente externamente usando o contexto da versão. A tabela a seguir mostra um exemplo de nomes que são identificados exclusivamente:

Tabela 16. Exemplo de Nomes e Acrônimos

Tipo de nome	Exemplo
Nome do Aplicativo de Processo	Aplicativo de Processo 1
Acrônimo do nome do aplicativo de processo	PA1
Captura instantânea do aplicativo de processo	1.0.0
Acrônimo da captura instantânea do aplicativo de processo	1.0.0

Um recurso, como um módulo ou uma biblioteca, tem o contexto de versão como parte de sua identificação.

A tabela a seguir mostra um exemplo de dois módulos e como os arquivos EAR associados incluem o contexto da versão:

Tabela 17. Módulos SCA e Arquivos EAR Reconhecidos pela Versão

Nome do módulo SCA	Nome Reconhecido pela Versão	Nome de EAR/Aplicativo Reconhecido pela Versão
M1	PA1-1.0.0-M1	PA1-1.0.0-M1.ear
M2	PA1-1.0.0-M2	PA1-1.0.0-M2.ear

A tabela a seguir mostra um exemplo de duas bibliotecas com escopo definido no aplicativo de processo e como os arquivos JAR associados incluem o contexto da versão:

Tabela 18. Bibliotecas com Escopo Definido no Aplicativo de Processo e Arquivos JAR Reconhecidos pela Versão

Nome da Biblioteca com Escopo Definido no Aplicativo de Processo SCA	Nome Reconhecido pela Versão	JAR Name Reconhecido pela Versão
Lib1	PA1-1.0.0-Lib1	PA1-1.0.0-Lib1.jar
Lib2	PA1-1.0.0-Lib2	PA1-1.0.0-Lib2.jar

Ligações Reconhecidas pela Versão

Os aplicativos de processo podem conter módulos SCA que incluem ligações de importação e exportação. Quando você implementar aplicativos conjuntamente, a ligação para cada versão do aplicativo deverá ser exclusiva. Algumas ligações são atualizadas automaticamente durante a implementação para assegurar a exclusividade entre versões. Em outros casos, você precisa atualizar a ligação após a implementação para garantir sua exclusividade.

Uma ligação *reconhecida pela versão* tem o escopo definido em uma versão específica de um aplicativo de processo, o que garante sua exclusividade entre aplicativos de processo. As seções a seguir descrevem as ligações que são atualizadas automaticamente para serem reconhecidas pela versão bem como quaisquer ações que você precisa executar no tempo de execução quando uma ligação não é reconhecida pela versão. Para obter informações sobre coisas a considerar quando você cria módulos, consulte “Considerações ao Usar Ligações”.

SCA

O destino de uma ligação de SCA é renomeado automaticamente para ser reconhecido pela versão durante a implementação se as ligações de importação e exportação do módulo são definidas no mesmo escopo do aplicativo de processo.

Se as ligações não estiverem definidas no mesmo escopo do aplicativo de processo, uma mensagem de informação será registrada. Você deve modificar a ligação de importação após a implementação para alterar o endereço de destino do terminal. É possível usar o console administrativo para alterar o endereço de destino do terminal.

Serviço da Web (JAX-WS ou JAX-RPC)

O endereço de destino de terminal de uma ligação de serviço da Web é automaticamente renomeado para ser reconhecido pela versão durante a implementação se todas as condições a seguir forem verdadeiras:

- Você seguiu as convenções de nomenclatura padrão para o endereço:

`http://ip:port/ModuleNameWeb/sca/ExportName`

- O endereço do terminal é SOAP/HTTP.
- As ligações de importação e exportação do módulo são definidas no mesmo escopo do aplicativo de processo.

Se estas condições não forem verdadeiras, uma mensagem de informação será registrada. A ação então executada dependerá de como você está implementando seu aplicativo de processo:

- Se estiver implementando conjuntamente seu aplicativo de processo, você deverá renomear manualmente a URL do terminal SOAP/HTTP ou a fila de destino SOAP/JMS para ser exclusiva entre as versões do aplicativo de processo. É possível usar o console administrativo após a implementação para alterar o endereço de destino do terminal.
- Se estiver implementando apenas uma única versão do aplicativo de processo, você poderá ignorar esta mensagem

Para implementação conjunta de captura instantânea de ligação de serviço da Web SOAP/JMS, a ação executada depende de como você está implementando seu aplicativo de processo:

- Se a importação e a exportação de destino estão no mesmo aplicativo de processo, execute as seguintes etapas antes de publicar o aplicativo de processo para o Process Center e criar a captura instantânea:
 1. Altere a URL do terminal da exportação. Certifique-se de que o destino e a connection factory sejam exclusivos.
 2. Altere a URL do terminal da importação para que seja igual à que você especificou para a exportação na etapa anterior.
- Se a importação e a exportação de destino estiverem em aplicativos de processo diferentes, execute as etapas a seguir:
 1. Altere a URL do terminal da exportação. Certifique-se de que o destino e a connection factory sejam exclusivos.
 2. Publique o aplicativo de processo no Process Center.
 3. Crie a captura instantânea.
 4. Implemente o aplicativo de processo no Process Server.
 5. Use o console administrativo do WebSphere para alterar a URL do terminal da importação correspondente para que seja igual à que você especificou para a exportação.

HTTP

O endereço da URL do terminal de uma ligação HTTP é renomeada automaticamente para ser reconhecida pela versão durante a implementação se todas as condições a seguir forem verdadeiras:

- Você seguiu as convenções de nomenclatura padrão para o endereço:
`http(s)://ip:port/ModuleNameWeb/contextPathInExport`
- As ligações de importação e exportação do módulo são definidas no mesmo escopo do aplicativo de processo.

Se estas condições não forem verdadeiras, uma mensagem de informação será registrada. A ação então executada dependerá de como você está implementando seu aplicativo de processo:

- Se estiver implementando conjuntamente seu aplicativo de processo, você deverá renomear manualmente a URL do terminal para ser exclusiva entre versões do aplicativo de processo. É possível usar o console administrativo após a implementação para alterar o endereço de destino do terminal.
- Se estiver implementando apenas uma única versão do aplicativo de processo, você poderá ignorar esta mensagem

JMS e JMS Genérico

Ligações JMS e JMS genérica geradas pelo sistema são automaticamente reconhecidas pela versão.

Nota: Para ligações JMS e JMS genéricas definidas pelo usuário, não ocorre nenhuma renomeação automática durante a implementação para ativar as ligações a serem reconhecidas pela versão. Se a ligação for definida pelo usuário, você deverá renomear os atributos a seguir para serem exclusivos entre versões do aplicativo de processo:

- Configuração de terminal
- Fila de destino de recebimento
- Nome da porta do listener (se definido)

Configure o destino de Envio correspondente se alterar o terminal do módulo de destino.

MQ/JMS e MQ

Nenhuma renomeação automática ocorre durante a implementação para permitir que ligações de tipo MQ/JMS ou MQ sejam reconhecidas pela versão.

Você deve renomear os seguintes atributos para serem exclusivos entre as versões do aplicativo de processo:

- Configuração de terminal
- Fila de destino de recebimento

Configure o destino de Envio correspondente se alterar o terminal do módulo de destino.

EJB

Não ocorre nenhuma renomeação automática durante a implementação para ativar ligações do tipo EJB para serem reconhecidas pela versão.

Você deve renomear o atributo de nome JNDI para ser exclusivo entre as versões do aplicativo de processo.

Observe que os aplicativos clientes também precisam ser atualizados para usar os novos nomes JNDI.

EIS

Um adaptador de recursos é automaticamente renomeado para ser reconhecido pela versão durante a implementação contanto que o nome do recurso padrão (*ModuleNameApp:Adapter Description*) não tenha sido modificado.

Se o nome do recurso padrão foi modificado, os nomes de adaptador de recursos deverão ser exclusivos entre as versões do aplicativo de processo.

Se os nomes de adaptador de recursos não forem exclusivos, uma mensagem informativa será registrada durante a implementação para alertá-lo. É possível renomear manualmente os adaptadores de recursos após a implementação usando o console administrativo.

Chamada Dinâmica Reconhecida pela Versão

É possível configurar componentes do fluxo de mediação para rotear mensagens para terminais que são determinados dinamicamente no tempo de execução. Quando você cria o módulo de mediação, configura a consulta de terminal para usar roteamento reconhecido pela versão.

Se você usar o estilo IBM_VRM (*<version>.<release>.<modification>*) para a captura instantânea, poderá exportar o arquivo EAR do aplicativo de processo para o WebSphere Service Registry and Repository (WSRR). Quando você cria o módulo de mediação, em seguida, configura a consulta de terminal para usar roteamento reconhecido pela versão. Por exemplo, você seleciona **Retornar versão compatível mais**

recente correspondente ao terminal dos serviços baseados em módulo SCA a partir do campo **Corresponder Política** e você seleciona **SCA** para o **Tipo de Ligação**.

Futuras versões do aplicativo de processo são implementadas no servidor e publicadas no WSRR e a consulta de terminal de módulo de mediação chama dinamicamente a versão compatível mais recente do terminal em serviço.

Observe que, como alternativa, você pode configurar o destino no SMOHeader e o valor pode ser transportado pela mensagem de solicitação.

Implementando Aplicativos de Processo com Módulos e Projetos Java

Aplicativos de processo podem conter módulos Java EE e projetos Java customizados. Quando você implementa aplicativos em conjunto, o módulo Java EE customizado para cada versão do aplicativo deve ser exclusivo.

Observe que os módulos Java EE e projetos Java customizados são implementados em um servidor se eles são implementados com um módulo SCA que possui uma dependência declarada deles. Se você não selecionar **Implementar com Módulo** (que é o padrão) quando declarar a dependência, será necessário implementar o módulo ou projeto manualmente.

Implementando Aplicativos de Processo com Regras de Negócios e Seletores

Se você estiver implementando diversas versões de um aplicativo de processo que inclui uma regra de negócios ou componente seletor, esteja ciente da maneira como os metadados associados são usados pelas versões.

Os metadados dinâmicos para uma regra de negócios ou componente seletor são definidos no tempo de execução pelo nome do componente, namespace de destino do componente e o tipo de componente. Se duas ou mais versões de um aplicativo de processo que contém uma regra de negócios ou seletor forem implementadas no mesmo ambiente de tempo de execução, elas compartilharão a mesma lógica de regra (regra de negócios) ou metadados de roteamento (seletor).

Para permitir que cada versão da regra de negócios ou componente seletor do aplicativo de processo use seus próprios metadados dinâmicos (lógica da regra ou roteamento), refatore o namespace de destino para que seja exclusivo para cada versão do aplicativo de processo.

Objetos de Configuração

É possível usar a ferramenta de administração de linha de comandos AdminConfig (wsadmin) do WebSphere para acessar e modificar as propriedades do banco de dados e de segurança no IBM Business Process Manager.

O termo *objeto de configuração* se refere a um objeto que é acessado usando os comandos wsadmin AdminConfig. Para obter mais informações, consulte Propriedades de Configuração de Segurança.

Arquitetura de Implementação

A arquitetura de implementação do IBM Business Process Manager consiste em processos de software chamados servidores, unidades topológicas referenciadas como nós e células, e o repositório de configuração usado para armazenar as informações de configuração.

Células

No IBM Business Process Manager, *Células* são agrupamentos lógicos de um ou mais nós em uma rede distribuída.

Uma célula é um conceito de configuração, uma forma de os administradores associar logicamente nós entre si. Os administradores definem os nós que compõem uma célula, de acordo com critérios específicos que fazem sentido em seus ambientes organizacionais.

Dados da configuração administrativa são armazenados em arquivos XML. Uma célula retém arquivos de configuração principal para cada servidor em cada nó na célula. Cada nó e servidor também têm seus próprios arquivos de configuração local. As alterações em um nó local ou em um arquivo de configuração de servidor serão temporárias, se o servidor pertencer à célula. Enquanto em efeito, as alterações locais substituem as configurações da célula. Alterações no servidor principal e nos arquivos de configuração do nó principal feitas no nível de célula substituem qualquer alteração temporária feita no nó quando os documentos de configuração da célula estão sincronizados nos nós. A sincronização ocorre em eventos designados, tal como quando um servidor inicia.

Servidores

Servidores fornecem a principal funcionalidade do IBM Business Process Manager. Os servidores de processo estendem ou aumentam a possibilidade de um servidor de aplicativos em manipular os módulos da Service Component Architecture (SCA). Outros servidores (gerenciadores de implementação e agentes de nó) são utilizados para gerenciar servidores de processo.

Um servidor de processo pode ser um *servidor independente* ou um *servidor gerenciado*. Um servidor gerenciado pode, opcionalmente, ser um membro de um *cluster*. Uma coleta de servidores gerenciados, clusters de servidores e outros tipos de middleware é chamada de *ambiente de implementação*. Em um ambiente de implementação, cada um dos servidores ou clusters gerenciados é configurado para uma função específica no ambiente de implementação (por exemplo, host de destino, host do módulo do aplicativo ou o servidor da Common Event Infrastructure). Um servidor independente é configurado para fornecer todas as funções necessárias.

Os servidores fornecem o ambiente de tempo de execução para os módulos SCA, para os recursos que são usados por esses módulos (origens de dados, especificações de ativação e destinos JMS), e para recursos fornecidos pela IBM (destinos de mensagens, contêineres do Business Process Choreographer e servidores da Common Event Infrastructure).

Um *agente de nó* é um agente administrativo que representa um nó para o seu sistema e gerencia os servidores nesse nó. Os agentes de nós monitoram os servidores em um sistema host e roteiam pedidos administrativos para os servidores. O agente do nó é criado quando um nó é associado a um gerenciador de implementação.

Um *gerenciador de implementação* é um agente administrativo que fornece uma visualização de gerenciamento centralizada para vários servidores e clusters.

Um servidor independente é definido pelo perfil independente; um gerenciador de implementação é definido por um perfil de gerenciador de implementação; servidores gerenciados são criados em um *nó gerenciado*, que está definido por um perfil gerenciado por nó.

Servidores Independentes:

Um servidor independente fornece um ambiente para implementar módulos de SCA em um processo do servidor. Este processo do servidor inclui, mas não se limita a, um console administrativo, um destino de implementação, o suporte ao sistema de mensagens, o gerenciador de regras de processo de negócios e um servidor de Common Event Infrastructure.

Um servidor independente é simples de configurar e possui um console de iniciação rápida do qual é possível iniciar e parar o servidor e abrir a galeria de amostras e o console administrativo. Se você instalar as amostras do IBM Business Process Manager e, em seguida, abrir a galeria de amostras, uma solução de amostra será implementada no servidor independente. Você pode explorar os recursos utilizados para esta amostra no console administrativo.

É possível implementar suas próprias soluções em um servidor independente, mas um servidor independente não pode fornecer a capacidade, a escalabilidade ou a robustez requeridas de um ambiente de produção. Para o seu ambiente de produção, é melhor utilizar um ambiente de implementação de rede.

É possível iniciar com um servidor independente e posteriormente incluí-lo em um ambiente de implementação de rede, federando-o para uma célula do gerenciador de implementação, *contanto que nenhum outro nó tenha sido federado nessa célula*. Não é possível federar vários servidores independentes em uma célula. Para federar o servidor independente, utilize o console administrativo do gerenciador de implementação ou o comando **addNode**. O servidor independente não deve estar em execução ao federá-lo utilizando o comando **addNode**.

Um servidor independente é definido por um perfil do servidor independente.

Clusters:

Os clusters são grupos de servidores que são gerenciados juntos e participam do gerenciamento de carga de trabalho.

Um cluster pode conter nós ou servidores de aplicativos individuais. Um nó geralmente é um sistema de computador físico com um endereço IP de host distinto que está executando um ou mais servidores de aplicativos. Clusters podem ser agrupados sob a configuração de uma célula, que associa logicamente muitos servidores e clusters com diferentes configurações e aplicativos um com o outro dependendo do critério do administrador e do que faz sentido em seus ambientes organizacionais.

Os clusters são responsáveis pelo equilíbrio de carga de trabalho entre os servidores. Servidores que são uma parte de um cluster são chamados de membros de cluster. Quando você instala um aplicativo em um cluster, o aplicativo é instalado automaticamente em cada membro de cluster.

Como cada membro de cluster contém os mesmos aplicativos, é possível distribuir tarefas do cliente em plataformas distribuídas de acordo com as capacidades das diferentes máquinas designando pesos a cada servidor.

Designar pesos aos servidores em um cluster melhora o desempenho e o failover. As tarefas são designadas aos servidores que têm a capacidade de executar as operações de tarefas. Se um servidor estiver disponível para executar a tarefa, ela será designada para outro membro do cluster. Este recurso de redesignação possui vantagens óbvias sobre a execução de um único servidor de aplicativos que pode se tornar sobrecarregado se muitas solicitações são feitas.

Perfis

Um perfil define um ambiente de tempo de execução exclusivo, com arquivos de comandos, arquivos de configuração e arquivos de log separados. Os perfis definem três tipos diferentes de ambientes em sistemas IBM Business Process Manager: servidor independente, gerenciador de implementação e nó gerenciado.

Ao usar perfis, é possível ter mais de um ambiente de tempo de execução em um sistema, sem ter de instalar cópias múltiplas dos arquivos binários do IBM Business Process Manager.

Use o Utilitário de Linha de Comandos BPMConfig para criar perfis do IBM BPM. O utilitário de linha de comandos **manageprofiles** ou sua interface gráfica com o usuário, o Profile Management Tool (PMT), pode ser usado como formas alternativas de criar o gerenciador de implementação ou perfis gerenciados por nós. O PMT não é mais suportado para criação de perfil independente.

Nota: Em plataformas distribuídas, cada perfil possui um nome exclusivo. No z/OS, todos os perfis são denominados “padrão”; não é possível renomear, editar, copiar ou excluir perfis no z/OS.

Tipos de Perfil

Os tipos de perfil a seguir do IBM BPM estão disponíveis com o IBM Business Process Manager V8.5:

Perfil independente do IBM BPM

O perfil independente define servidores independentes com recursos e funcionalidade específicos às configurações do IBM BPM Express. É possível criar o perfil independente usando o modelo de perfil BPM/BpmServer, que é suportado somente no IBM BPM Express.

Perfil de gerenciador de implementação do IBM BPM

O perfil de gerenciador de implementação define um gerenciador de implementação que fornece uma interface administrativa para um grupo lógico de servidores em uma ou mais estações de trabalho. É possível criar o perfil de gerenciador de implementação usando o modelo de perfil BPM/BpmMgr, que é suportado somente no IBM BPM Standard e IBM BPM Advanced.

Perfil gerenciado por nó do IBM BPM

O perfil gerenciado por nó define um nó gerenciado quando nó é federado para um gerenciador de implementação. É possível criar o perfil gerenciado por nó usando o modelo de perfil BPM/BpmNode, que é suportado somente no IBM BPM Standard e IBM BPM Advanced.

Os tipos de perfil específicos estão disponíveis com cada configuração do IBM BPM.

Tabela 19. Tipos de Perfil Disponíveis do IBM BPM

Configuração do IBM BPM	Tipos de Perfil		
	Independente	Gerenciador de Implementação	Gerenciado por nó
IBM BPM Expresso	Sim	No	No
IBM BPM Standard	No	Sim	Sim
IBM BPM Advanced	No	Sim	Sim
Unit Test Environment (UTE) for IBM Integration Designer	Sim	Opcional	Opcional

O Diretório do Perfil

Cada perfil no sistema tem seu próprio diretório, contendo todos seus arquivos. O local do diretório de perfil é especificado ao criar o perfil. Por padrão, ele se encontra no diretório `profiles`, no diretório no qual o IBM Business Process Manager está instalado. Por exemplo, o perfil `Dmgr` encontra-se em `C:\Program Files\IBM\WebSphere\ProcServer\profiles\Dmgr`.

O Perfil Padrão

O primeiro perfil criado dentro de uma instalação do IBM Business Process Manager é o *perfil padrão*. O perfil padrão é o destino padrão para comandos emitidos a partir do diretório `bin` no diretório em que o IBM Business Process Manager foi instalado. Se existir apenas um perfil em um sistema, todos os comandos operarão nesse perfil. Se você criar um outro perfil, você poderá torná-lo o padrão.

Nota: O perfil padrão não é necessariamente um perfil cujo nome é "default".

Aprimorando Perfis

Se já tiver um perfil de gerenciador de implementação, um perfil gerenciado por nó ou um perfil do servidor independente criado para WebSphere Application Server Network Deployment, é possível *umentá-lo* para suportar o IBM Business Process Manager além da função existente. Para aprimorar um perfil, primeiro instale o IBM Business Process Manager. Em seguida, use o utilitário de linha de

comandos **manageprofiles** para aumentar um perfil independente ou use o Profile Management Tool ou o utilitário de linha de comandos **manageprofiles** para aumentar um gerenciador de implementação ou um perfil gerenciado por nó.

Importante: Em um ambiente de implementação de rede, você precisa aumentar o perfil de gerenciador de implementação primeiro e depois aumentar os perfis de nó gerenciado.

Restrição: Não é possível aumentar um perfil de gerenciador independente ou de implementação onde o registro do usuário do WebSphere VMM padrão foi alterado, por exemplo, para usar o LDAP.

Gerenciadores de implementação

Um gerenciador de implementação é um servidor que gerencia operações para um grupo lógico, ou célula, de outros servidores. O gerenciador de implementação é o local central para administração de servidores e clusters.

Ao criar um ambiente de implementação, o perfil do gerenciador de implementação é o primeiro perfil a ser criado. Cada ambiente de implementação criado possui um console de iniciação rápida, a partir do qual é possível iniciar e parar o gerenciador de implementação e iniciar seu console administrativo. Utilize o console administrativo do gerenciador de implementação para gerenciar os servidores e clusters na célula. Isso inclui configurar servidores e clusters, incluir servidores nos clusters, iniciar e parar servidores e clusters e implementar os módulos SCA.

Embora o gerenciador de implementação seja um tipo de servidor, não é possível implementar módulos no próprio gerenciador de implementação.

Nós

Um *nó* é um agrupamento lógico de servidores gerenciados.

Um nó geralmente corresponde a um sistema de computador lógico ou físico com um endereço de host IP diferente. Os nós não podem abranger múltiplos computadores. Os nós de nomes geralmente são idênticos ao nome do host do computador.

Os nós na topologia de implementação de rede podem ser gerenciados ou não gerenciados. Um nó gerenciado tem um processo de agente do nó que gerencia sua configuração e servidores. Os nós não gerenciados não têm um agente do nó.

Nós Gerenciados:

Um *nó gerenciado* é um nó que está associado a um gerenciador de implementação e que contém um agente do nó, podendo conter servidores gerenciados. Em um nó gerenciado, você pode configurar e executar servidores gerenciados.

Os servidores que são configurados em um nó gerenciado, formam os recursos de seu ambiente de implementação. Estes servidores são criados, configurados, iniciados, parados, gerenciados e excluídos utilizando o console administrativo do gerenciador de implementação.

Um nó gerenciado contém um agente do nó que gerencia todos os servidores em um nó.

Quando um nó é federado, um processo do agente do nó é criado automaticamente. Este agente do nó deve estar em execução para ser capaz de gerenciar a configuração do perfil. Por exemplo, quando você realiza as seguintes tarefas:

- Iniciar e parar processos do servidor.
- Sincronizar dados de configuração no gerenciador de implementação com a cópia no nó.

Entretanto, o agente do nó não precisa estar em execução para que os aplicativos sejam executados ou para configurar recursos no nó.

Um nó gerenciado pode conter um ou mais servidores, que são gerenciados por um gerenciador de implementação. É possível implementar soluções para os servidores em um nó gerenciado, mas o nó gerenciado não contém uma galeria de aplicativos de amostras. O nó gerenciado é definido por um perfil gerenciado por nó e possui um console de Iniciação Rápida.

Nós não gerenciados:

Um nó não gerenciado não tem um agente de nó para gerenciar seus servidores.

Os nós não gerenciados na topologia de Implementação de Rede podem ter definições de servidor, como servidores da web, mas não definições de servidor de aplicativos. Os nós não gerenciados nunca podem ser federados. Ou seja, um agente do nó nunca pode ser incluído em um nó não gerenciado. Um outro tipo de nó não gerenciado é um servidor independente. O gerenciador de implementação não pode gerenciar este servidor independente porque ele não é conhecido na célula. Um servidor independente pode ser federado. Quando ele é federado, um agente do nó é criado automaticamente. O nó se torna um nó gerenciado na célula.

Agentes do nó

Os agentes do nó são agentes administrativos que roteiam pedidos administrativos para servidores.

Um agente do nó é um servidor que é executado em cada sistema do computador host que participa na configuração do Network Deployment. Ele é puramente um agente administrativo e não está envolvido nas funções de serviço do aplicativo. Um agente do nó também hospeda outras funções administrativas importantes tais como serviços de transferência de arquivos, sincronização de configuração e monitoramento de desempenho.

Considerações sobre a Nomenclatura de Perfis, Nós, Servidores, Hosts e Células

Este tópico discute os problemas e os termos reservados que deverão ser considerados ao nomear o perfil, o nó, o servidor, o host e a célula (se aplicável). Este tópico se aplica apenas às plataformas distribuídas.

Considerações de Nomenclatura do Perfil

O nome do perfil pode ser qualquer nome exclusivo com as restrições a seguir. Não utilize nenhum dos caracteres a seguir ao nomear seu perfil:

- Espaços
- Caracteres especiais que não são permitidos no nome de um diretório do sistema operacional, tal como *, & ou ?.
- Barras (/) ou barras invertidas (\)

Caracteres de byte duplo são permitidos.

Considerações sobre o caminho do diretório: O caminho do diretório de instalação deve ser menor ou igual a 60 caracteres. O número de caracteres no diretório *profiles_directory_path\profile_name* deve ser menor ou igual a 80 caracteres.

Nota: Use uma convenção de nomenclatura de caminho curto ao criar um perfil em um ambiente Windows para evitar a limitação de comprimento de caminho de 255 caracteres do Windows.

Considerações sobre Nomenclatura de Nó, Servidor, Host e Célula

Nomes reservados: Evite usar nomes reservados como valores de campos. O uso de nomes reservados pode causar resultados imprevisíveis. As seguintes palavras são reservadas:

- cells
- nodes

- servers
- clusters
- aplicativos
- deployments

Descrições dos campos nas páginas Nó e Nomes de Host e Nó, Host e Nomes de Célula: Use as diretrizes de nomenclatura apropriadas ao criar perfis.

- Perfis de servidores independentes
- Perfis do Gerenciador de Implementação
- Perfis gerenciados por nó

Tabela 20. Diretrizes de Nomenclatura para Perfis de Servidor Independente

Nome do Campo	Valor Padrão	Restrições	Descrição
Nome do nó	... <i>shortHostName</i> Node <i>NodeNumber</i> em que: <ul style="list-style-type: none"> • <i>shortHost Name</i> é o nome abreviado do host. • <i>NodeNumber</i> é um número sequencial que começa em 01. 	Evite utilizar os nomes reservados.	Selecione qualquer nome desejado. Para ajudar a organizar sua instalação, utilize um nome exclusivo se você planeja criar mais de um servidor no sistema.
Nome do servidor	... server1	Use um nome exclusivo para o servidor.	O nome lógico para o servidor.
Nome do host	... O formato longo do nome do DNS (Servidor de Nomes de Domínio).	Use um nome completo do host que seja endereçável por meio da sua rede.	Utilize o nome de DNS real ou o endereço IP de sua estação de trabalho para ativar a comunicação com ele. Consulte informações adicionais sobre o nome do host após esta tabela.

Tabela 21. Diretrizes de Nomenclatura para Perfis de Gerenciador de Implementação

Nome do Campo	Valor Padrão	Restrições	Descrição
Nome do nó	... <i>shortHostName</i> Cell <i>ManagerNode Number</i> em que: <ul style="list-style-type: none"> • <i>shortHost Name</i> é o nome abreviado do host. • <i>NodeNumber</i> é um número sequencial que começa em 01. 	Utilize um nome exclusivo para o gerenciador de implementação. Evite utilizar os nomes reservados.	O nome é utilizado para administração na célula do gerenciador de implementação.
Nome do host	... O formato longo do nome do DNS (Servidor de Nomes de Domínio).	Use um nome completo do host que seja endereçável por meio da sua rede. Evite utilizar os nomes reservados.	Utilize o nome de DNS real ou o endereço IP de sua estação de trabalho para ativar a comunicação com ele. Consulte informações adicionais sobre o nome do host após esta tabela.

Tabela 21. Diretrizes de Nomenclatura para Perfis de Gerenciador de Implementação (continuação)

Nome do Campo	Valor Padrão	Restrições	Descrição
Nome da Célula	<p>... <i>shortHostName</i> Cell <i>CellNumber</i> em que:</p> <ul style="list-style-type: none"> • <i>shortHost Name</i> é o nome abreviado do host. • <i>CellNumber</i> é um número sequencial que começa em 01. 	<p>Utilize um nome exclusivo para a célula do gerenciador de implementação. Um nome de célula deve ser exclusivo em qualquer circunstância na qual o produto está em execução na mesma estação de trabalho física ou cluster de estações de trabalho, como um Sysplex. Além disso, um nome de célula deve ser exclusivo em qualquer circunstância na qual a conectividade de rede entre entidades é requerida, entre as células ou a partir de um cliente que deve comunicar-se com cada uma das células. Os nomes de células também deverão ser exclusivos se seus espaços de nomes tiverem que ser associados. Caso contrário, você pode encontrar sintomas, tais como uma exceção <code>javax.naming.NameNotFoundException</code> que, nesse caso, é necessário criar células exclusivamente denominadas.</p>	<p>Todos os nós federados tornam-se membros da célula do gerenciador de implementação, que é nomeada na página Nomes de Nós, Hosts e Células do Profile Management Tool.</p>

Tabela 22. Diretrizes de Nomenclatura para Perfis Gerenciados por Nó

Nome do Campo	Valor Padrão	Restrições	Descrição
Nome do nó	<p>... <i>shortHostName</i> Node <i>NodeNumber</i> em que:</p> <ul style="list-style-type: none"> • <i>shortHost Name</i> é o nome abreviado do host. • <i>NodeNumber</i> é um número sequencial que começa em 01. 	<p>Evite utilizar os nomes reservados.</p> <p>Utilize um nome exclusivo na célula do gerenciador de implementação.</p>	<p>O nome é usado para administração na célula do gerenciador de implementação à qual o perfil gerenciado por nó está incluída. Utilize um nome exclusivo na célula do gerenciador de implementação.</p>
Nome do host	<p>... O formato longo do nome do DNS (Servidor de Nomes de Domínio).</p>	<p>Use um nome completo do host que seja endereçável por meio da sua rede.</p>	<p>Utilize o nome de DNS real ou o endereço IP de sua estação de trabalho para ativar a comunicação com ele. Consulte informações adicionais sobre o nome do host após esta tabela.</p>

Considerações sobre o Nome do Host:

O nome do host é o nome da rede para a estação de trabalho física na qual o nó está instalado. O nome do host deve indicar um nó de rede física no servidor. Quando há várias placas de rede no servidor, o nome do host ou o endereço IP deve indicar uma das placas de rede. Os nós remotos utilizam o nome do host para conexão e comunicação com o nó.

O IBM Business Process Manager é compatível com o Internet Protocol versão 4 (IPv4) e versão 6 (IPv6). Onde quer que você possa digitar endereços IP no console administrativo, ou em outro lugar qualquer, você pode fazê-lo em qualquer um dos formatos. Observe que, se o IPv6 estiver implementado em seu sistema, você deve digitar o endereço IP no formato IPv6, contudo, se o IPv6 ainda não estiver disponível, digite endereços IP no formato IPv4. Para obter informações adicionais sobre o IPv6 consulte a descrição a seguir : IPv6.

As seguintes diretrizes podem ajudar na determinação do nome de host apropriado para sua estação de trabalho:

- Selecione um nome de host que outras estações de trabalho possam acessar em sua rede.
- Não utilize o identificador genérico, host local, para este valor.
- Não tente instalar produtos do IBM Business Process Manager em um servidor com um nome do host que utiliza caracteres de DBCS (Conjunto de Caracteres de Byte Duplo). Caracteres DBCS não são suportados quando utilizados no nome do host.
- Evite utilizar o caractere sublinhado (_) nos nomes de servidor. Padrões da Internet declaram que nomes de domínio devem estar conformidade com os requisitos de nome do host descritos nos padrões Internet Official Protocol Standards RFC 952 e RFC 1123. Os nomes de domínio devem conter somente letras (maiúsculas ou minúsculas) e dígitos. Os nomes de domínio também podem conter caracteres de traço (-) contanto que os traços não estejam nas extremidades do nome. Os caracteres sublinhados (_) não são suportados no nome do host. Se você tiver instalado o IBM Business Process Manager em um servidor com um caractere sublinhado no nome do servidor, acesse o servidor com seu endereço IP até renomeá-lo.

Se você definir nós coexistentes no mesmo computador com endereços IP exclusivos, defina cada endereço IP em uma tabela de consulta de DNS (Domain Name Server). Os arquivos de configuração para servidores não fornecem a resolução de nome de domínio para endereços IP múltiplos em uma estação de trabalho com um único endereço de rede.

O valor que você especifica para o nome do host é utilizado como o valor da propriedade hostName nos documentos de configuração. Especifique o valor do nome do host em um dos seguintes formatos:

- Cadeia do nome completo do host DNS (Domain Name Servers), como xmachine.manhattan.ibm.com
- A cadeia do nome abreviado do host DNS padrão, como xmachine
- Endereço IP numérico, como 127.1.255.3

O nome completo do host DNS tem as vantagens de ser totalmente inequívoco e flexível. Você tem a flexibilidade de alterar o endereço IP real do sistema host sem ter de alterar a configuração do servidor. Esse valor para o nome do host é especialmente útil se você pretende alterar o endereço IP com frequência ao utilizar o DHCP (Dynamic Host Configuration Protocol) para atribuir endereços IP. Uma desvantagem desse formato é ser dependente do DNS. Se o DNS não estiver disponível, a conectividade ficará comprometida.

O nome abreviado do host também pode ser resolvido dinamicamente. Um formato de nome abreviado tem a capacidade adicional de ser redefinido no arquivo de hosts locais para que o sistema possa executar o servidor mesmo quando desconectado da rede. Defina o nome abreviado para 127.0.0.1 (auto-retorno local) nos arquivos de hosts para executar desconectado. Uma desvantagem do formato de nome abreviado é ser dependente do DNS para acesso remoto. Se o DNS não estiver disponível, a conectividade ficará comprometida.

Um endereço IP numérico tem a vantagem de não requerer uma resolução de nome por meio do DNS. Um nó remoto pode conectar-se ao nó denominado com um endereço IP numérico sem o DNS estar disponível. Uma desvantagem desse formato é o endereço IP numérico ser fixo. Você deve alterar a configuração da propriedade `hostName` nos documentos de configuração sempre que alterar o endereço IP da estação de trabalho. Portanto, não utilize um endereço IP numérico se você utilizar DHCP ou se você alterar os endereços IP regularmente. Outra desvantagem desse formato é não ser possível usar o nó se o host estiver desconectado da rede.

BPMN 2.0

As definições do processo de negócio do IBM Business Process Manager suportam a subclasse Executável Comum da classe de conformidade Modelagem de Processo do BPMN 2.0, que lida com modelos que podem ser executados.

O Business Process Model and Notation (BPMN) é o padrão fundamental para os processos no IBM Process Designer e no IBM Process Center. Os diagramas de Business Process Definition (BPD) baseiam-se na especificação do BPMN. Esse tópico apresenta algumas maneiras pelas quais o BPMN 2.0 é aplicado no IBM Business Process Manager. Para obter informações detalhadas sobre o BPMN, consulte a página de Especificação do BPMN em <http://www.bpmn.org/>.

O IBM Business Process Manager suporta os seguintes tipos de tarefas do BPMN 2.0:

- Nenhum (tarefa abstrata na especificação do BPMN 2.0)
- Tarefa do sistema (tarefa de serviço na especificação do BPMN 2.0)
- Tarefa do usuário
- Script
- Tarefa de decisão (tarefa de regras de negócio na especificação do BPMN 2.0)

Os eventos de mensagem intermediária do IBM BPM fornecem funções similares à tarefa de envio e tarefa de recebimento do BPMN.

Notação do BPMN 2.0

A partir do V7.5.1, os ícones de tarefa do Process Designer BPMN 2.0 nos diagramas do BPD são coletados em uma paleta simplificada e exibidos nos diagramas de processo. Os ícones mostram se sua atividade é uma tarefa do sistema, tarefa do usuário, tarefa de decisão, script ou processo vinculado. As atividades em modelos que foram criados em versões anteriores também mostram tipos de tarefas e ícones de tarefas apropriadas do BPMN 2.0 ao visualizá-los na versão 7.5.1 ou posterior.

Atividades e Tarefas

Existem algumas mudanças de terminologia das versões anteriores do Process Designer. Algumas dessas mudanças envolvem tipos de atividade que foram renomeados.

- Atividades de serviço (automatizadas) agora são tarefas do sistema.
- Atividades de serviço (tarefa) em uma raia que não pertence ao sistema agora são tarefas do usuário.
- Atividades de serviço (tarefa) em uma raia do sistema agora são tarefas de decisão se fizerem referência a um serviço de decisão.
- Atividades de serviço (tarefa) em uma raia do sistema agora são tarefas do sistema se fizerem referência a qualquer tipo de serviço que não seja um serviço de decisão.
- Atividades Javascript agora são tarefas de script.
- Atividades de processo aninhadas agora são processos vinculados.
- Atividades externas de versões anteriores do Process Designer estão disponíveis como implementações externas para tarefas de usuários ou tarefas do sistema.

Gateways

Não há nenhuma mudança de notação para os gateways de versões anteriores. Entretanto, há três mudanças de terminologia. O gateway de decisão agora é *gateway exclusivo*, o gateway de divisão ou junção simples agora é *gateway paralelo*, e o gateway de divisão ou de junção condicional agora é *gateway inclusivo*.

Também há um novo tipo de gateway, o *gateway de evento*. Um gateway de evento representa um ponto de ramificação em um processo no qual os caminhos alternativos que seguem o gateway se baseiam nos eventos que ocorrem, em vez da avaliação das expressões usando dados do processo (assim como um gateway exclusivo ou inclusivo). Um evento específico, geralmente o recebimento de uma mensagem, determina o caminho que será usado.

Eventos de Não Interrupção

Notação incluída do BPMN 2.0 para eventos de não interrupção. Por padrão, um evento de limite interrompe a atividade à qual está anexada. Quando o evento é acionado, a atividade para e o token continua com o fluxo de sequência de saída do evento. Se o evento está configurado como não interrupção, quando o evento for acionado, a atividade anexada continua em paralelo e um novo token é gerado e passado para o fluxo de sequência de saída do evento. O limite de evento se altera para uma linha tracejada para os eventos de não interrupção.

Eventos intermediários que foram anexados em atividades são eventos intermediários de interrupção se fecharem suas atividades anexadas ou eventos intermediários de não interrupção se não fecharem suas atividades anexadas.

Iniciar evento

A especificação BPMN permite que os modelos de processo omitam símbolos de início e término de evento. O Process Designer requer que os modelos de processo usem eventos de início e término.

Há vários tipos de eventos de inícios disponíveis no Process Designer:

processos

- nenhum
- Mensagem
- ad hoc

subprocessos

- nenhum

subprocessos de evento

- erro
- Mensagem
- cronômetro

É possível alterar o tipo de um evento de início ao editar as propriedades do evento. É possível ter diversos eventos de início de mensagem em um processo, mas é possível usar somente um evento sem início.

Eventos de Término

Quatro tipos de eventos de terminos estão disponíveis: *mensagem*, *finalizar*, *erro* e *nenhum*. É possível alterar o tipo de um evento de término.

Quando um processo-pai chama um processo-filho e o processo-filho executa uma ação de evento de finalização, o processo-filho para e o processo-pai continua então com suas próximas etapas.

Subprocessos

A especificação BPMN define dois tipos de subprocessos, integrados e reutilizáveis. É possível criar ambos os tipos no Process Designer. Os subprocessos integrados são chamados apenas de *subprocessos* no Process Designer e na nova versão 7.5.1. O subprocesso reutilizável do BPMN é chamado de um *processo vinculado* no Process Designer.

Um subprocesso existe no processo contido e é uma forma de agrupar etapas de processo para reduzir a complexidade e a confusão do diagrama. Subprocessos reduzem diversas etapas a uma atividade. O subprocesso pode ser visto somente pelo processo no qual está definido. Um subprocesso existe no escopo de seu responsável pela chamada e possui acesso a todas as variáveis nesse ambiente. Nenhum parâmetro entra e sai do subprocesso integrado.

Separado do subprocesso e do processo vinculado, o Process Designer possui um subprocesso de evento, que é um subprocesso especializado que é usado para manipulação de eventos. Ele não é conectado as outras atividades por meio de fluxo de sequência, e ele ocorre somente se seu evento de início for acionado.

Processos Vinculados

Um subprocesso reutilizável do BPMN é chamado de *processo vinculado* no Process Designer. É um processo criado fora do processo atual que pode ser chamado pelo processo atual. Ele é reutilizável porque outras definições de processo também podem chamar esse processo. O processo vinculado define seus parâmetros de entrada e saída e não possui acesso ao escopo ou ambiente do responsável pela chamada. O processo vinculado é semelhante ao processo aninhado disponível em versões anteriores; não há mudança no comportamento da atividade. Processos aninhados anteriores são migrados para processos vinculados. O processo vinculado é semelhante a um subprocesso com um limite espesso e é destacado na janela do Inspector.

Loops

O BPMN fornece o conceito de uma atividade que pode ser repetida. A atividade pode ser atômica, o que significa que a atividade se repete, ou ela pode ser um subprocesso, encapsulando uma série de etapas que são repetidas. Se expandir a atividade repetida, verá as atividades contidas que devem ser executadas repetidamente. A condição é sempre avaliada no início de cada iteração de loop. Não há nenhuma possibilidade de avaliar o final de cada iteração de loop.

O IBM Business Process Manager possui um *loop multi-instância*, que é executado um número finito de vezes com as atividades contidas na execução sequencialmente ou em paralelo.

Importando Processos Não BPMN

É possível importar modelos que foram criados no IBM WebSphere Business Modeler e usá-los no Process Designer. Para obter informações sobre a importação do BPMN 2.0, consulte Mapeando elementos do IBM WebSphere Business Modeler para construções do IBM Business Process Manager. Também é possível importar modelos do BPMN 2.0 que foram criados no IBM WebSphere Business Compass, Rational Software Architect ou outros ambientes de modelagem.

Business Process Definitions (BPDs)

Para modelar um processo no IBM Process Designer, você deve primeiro criar uma Business Process Definition (BPD). A definição de processo de negócios pode basear-se em um modelo BPMN importado.

Uma Business Process Definition é um modelo reutilizável de um processo que define o que é comum a todas as instâncias de tempo de execução desse modelo de processo. Uma BPD deve conter um evento de início, um evento de término e pelo menos uma pista, e uma ou mais atividades. Consulte "Convenções de nomenclatura do Designer de Processo IBM" nos links relacionados para obter detalhes sobre as limitações de caractere que se aplicam às BPDs.

Uma Business Process Definition (BPD) precisa incluir uma pista para cada sistema ou grupo de usuários que participa em um processo. Uma pista pode ser uma pista participante ou uma pista do sistema. Porém, é possível criar uma BPD que agrupe as atividades de um grupo e um sistema em uma única pista se essa for a sua preferência. Consulte "Criando uma Business Process Definition (BPD)" nos links relacionados para informações sobre como criar uma BPD.

É possível designar qualquer pessoa ou grupo específico para ser responsável pelas atividades em uma pista participante. Cada pista criada é designada ao grupo Todos os Usuários por padrão. É possível usar este grupo padrão para execução e teste do seu BPD no Inspector. O grupo Todos os Usuários inclui todos os usuários que são membros do grupo de segurança `tw_allusers`, que é um grupo de segurança especial que inclui automaticamente todos os usuários no sistema.

Uma pista do sistema contém atividades manipuladas por um sistema IBM Process Center específico. Cada atividade precisa de uma implementação, que define a atividade e configura as propriedades para a tarefa. Durante a implementação, um desenvolvedor cria um serviço ou grava o JavaScript necessário para concluir as atividades em uma pista do sistema. Consulte "Entendendo tipos de serviço" nos links relacionados para informações sobre serviços.

Para cada BPD que criar, você precisará declarar variáveis para capturar os dados de negócios que são passados de atividade para atividade no seu processo. Consulte "Gerenciando e Mapeando Variáveis" nos links relacionados para aprender sobre implementação de variáveis.

Também é possível incluir eventos em uma BPD. Eventos em uma definição de processo de negócios IBM podem ser acionados pela passagem de um prazo final, uma exceção, ou a chegada de uma mensagem. O acionador que você deseja determina o tipo de evento que você escolhe para implementar. Para informações detalhadas sobre tipos de evento disponíveis e seus acionadores, consulte "Modelando eventos".

Ligações

No núcleo de uma arquitetura orientada a serviços está o conceito de um *serviço*, uma unidade de funcionalidade executada por uma interação entre dispositivos de computação. Uma *exportação* define a interface externa (ou ponto de acesso) de um módulo, de forma que os componentes Service Component Architecture (SCA) dentro do módulo possam fornecer seus serviços para clientes externos. Uma *importação* define uma interface para serviços fora de um módulo, de forma que os serviços possam ser chamados a partir de dentro do módulo. Você pode usar *ligações* específicas do protocolo com importações e exportações para especificar os meios de transportar os dados para dentro ou para fora do módulo.

Exportações

Clientes externos podem chamar os componentes SCA em um módulo de integração por meio de vários protocolos (como HTTP, JMS, MQ e RMI/IIOP) com dados em uma variedade de formatos (como XML, CSV, COBOL e JavaBeans). As exportações são componentes que recebem essas solicitações de origens externas e, em seguida, chamam componentes do IBM Business Process Manager usando o modelo de programação SCA.

Por exemplo, na figura a seguir, uma exportação recebe uma solicitação sobre o protocolo HTTP a partir de um aplicativo cliente. Os dados são transformados em um objeto de negócios, o formato usado pelo

componente SCA. O componente é, então, chamado com esse objeto de dados.

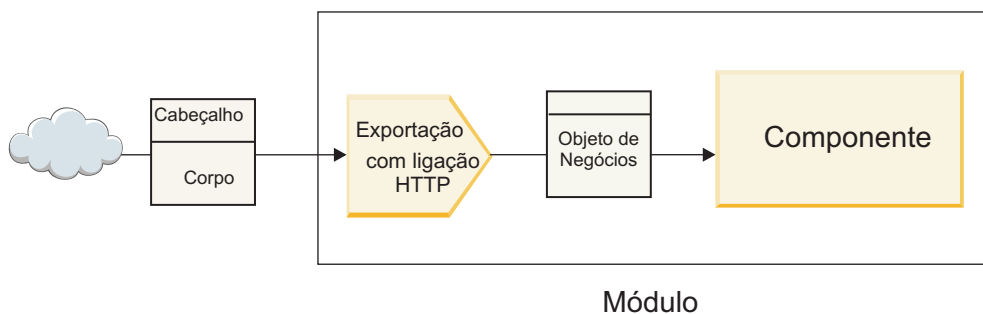


Figura 1. Uma Exportação com Ligação HTTP

Importações

Um componente SCA pode desejar chamar um serviço externo não SCA que espera dados em um formato diferente. Uma importação é usada pelo componente SCA para chamar o serviço externo usando o modelo de programação SCA. A importação, então, chama o serviço de destino da maneira que o serviço espera.

Por exemplo, na figura a seguir, uma solicitação a partir de um componente SCA é enviada, pela importação, para um serviço externo. O objeto de negócios, que é o formato usado pelo componente SCA, é transformado no formato esperado pelo serviço e o serviço é chamado.

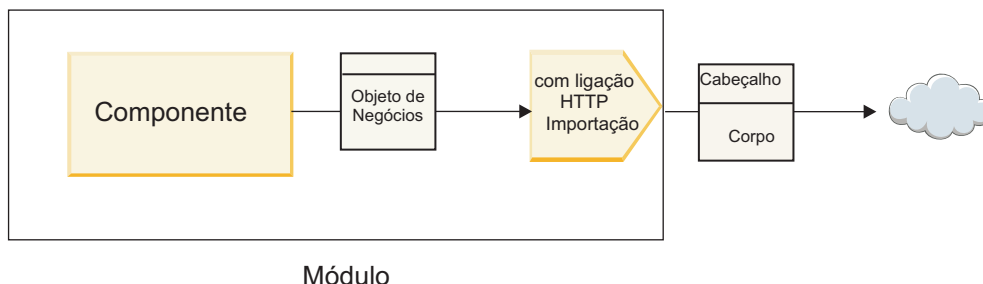


Figura 2. Uma Importação com Ligação HTTP

Lista de Ligações

Use o Integration Designer para gerar uma ligação para uma importação ou exportação e para configurar a ligação. Os tipos de ligações que estão disponíveis são descritos na lista a seguir.

- SCA

A ligação SCA, que é o padrão, permite que seu serviço se comunique com serviços em outros módulos SCA. É possível usar uma importação com uma ligação SCA para acessar um serviço em um outro módulo SCA. Use uma exportação com uma ligação SCA para oferecer um serviço para outros módulos SCA.

- Serviço da Web

Uma ligação de serviço da web permite acessar um serviço externo usando mensagens SOAP interoperáveis e qualidades de serviço. Também é possível usar ligações de serviço da web para incluir anexos como parte da mensagem SOAP.

A ligação de serviço da web pode usar um protocolo de transporte de SOAP/HTTP (SOAP sobre HTTP) ou SOAP/JMS (SOAP sobre JMS). Independentemente do transporte (HTTP ou JMS) usado para transmitir as mensagens SOAP, as ligações de serviço da web sempre manipulam interações de solicitação/resposta de forma síncrona.

- HTTP

A ligação HTTP permite que você acesse um serviço externo usando o protocolo HTTP, onde as mensagens não SOAP são usadas ou onde o acesso HTTP direto é necessário. Esta ligação é usada ao trabalhar com serviços da web baseados no modelo de HTTP (isto é, serviços que usam operações de interface HTTP bem conhecidas como GET, PUT, DELETE e assim por diante).

- Enterprise JavaBeans (EJB)

As ligações EJB permitem que componentes SCA interajam com os serviços fornecidos pela lógica de negócios Java EE em execução em um servidor Java EE.

- EIS

A ligação EIS (Enterprise Information System), quando usada com um adaptador de recursos JCA, permite acessar serviços em um Enterprise Information System ou tornar seus serviços disponíveis para o EIS.

- Ligações JMS

As ligações Java Message Service (JMS), JMS genéricas e JMS do WebSphere MQ (JMS do MQ) são usadas para interações com sistemas de mensagens, onde a comunicação assíncrona por meio das filas de mensagens é crítica para confiabilidade.

Uma exportação com uma das ligações JMS observa uma fila para a chegada de uma mensagem e envia de forma assíncrona a resposta, se houver, para a fila de resposta. Uma importação com uma das ligações JMS constrói e envia uma mensagem para uma fila JMS e observa uma fila para a chegada da resposta, se houver.

- JMS

A ligação JMS permite acessar o o provedor JMS integrado pelo WebSphere.

- JMS Genérico

A ligação JMS genérica permite acessar um sistema de mensagens do fornecedor não IBM.

- MQ JMS

A ligação JMS do MQ permite acessar o subconjunto JMS de um sistema de mensagens WebSphere MQ. Você usaria esta ligação quando o subconjunto JMS de funções é suficiente para seu aplicativo.

- MQ

A ligação do WebSphere MQ permite que você se comunique com aplicativos nativos do MQ, trazendo-os para a estrutura de arquitetura orientada a serviços e fornecendo acesso às informações de cabeçalho específicas do MQ. Você usaria esta ligação quando precisasse usar funções nativas do MQ.

Visão Geral da Ligação de Exportação e Importação

Uma exportação permite tornar os serviços em um módulo de integração disponíveis para clientes externos e uma importação torna possível para seus componentes SCA em um módulo de integração chamar serviços externos. A ligação associada à exportação ou importação especifica o relacionamento entre as mensagens de protocolo e os objetos de negócios. Ela também especifica a forma com que as operações e falhas são selecionadas.

Fluxo de Informações por meio de uma Exportação

Uma exportação recebe uma solicitação, que é destinada para o componente ao qual a exportação está ligada, por meio de um transporte específico determinado pela ligação associada (por exemplo, HTTP).

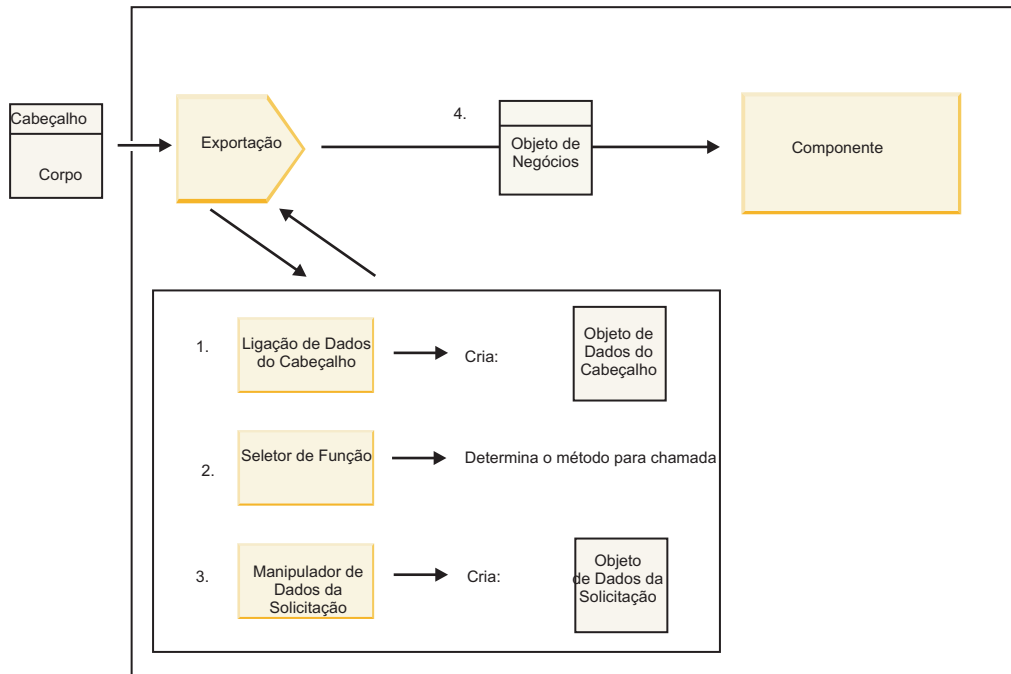


Figura 3. Fluxo de uma Solicitação por meio da Exportação para um Componente

Quando a exportação recebe a solicitação, a seguinte sequência de eventos ocorre:

1. Para ligações do WebSphere MQ apenas, a ligação de dados do cabeçalho transforma o cabeçalho do protocolo em um objeto de dados do cabeçalho.
2. O seletor de função determina o nome do método nativo da mensagem de protocolo. O nome do método nativo é mapeado pela configuração de exportação para o nome de uma operação na interface da exportação.
3. O manipulador de dados da solicitação ou a ligação de dados no método transforma a solicitação em um objeto de negócios de solicitação.
4. A exportação chama o método do componente com o objeto de negócios de solicitação.
 - A ligação de exportação HTTP, a ligação de exportação de serviço da web e a ligação de exportação EJB chamam o componente SCA de forma síncrona.
 - As ligações de exportação JMS, JMS Genéricas, JMS do MQ e do WebSphere MQ chamam o componente SCA de forma assíncrona.

Observe que uma exportação pode propagar os cabeçalhos e as propriedades do usuário que ela recebe por meio do protocolo, se a propagação do contexto está ativada. Componentes que estão ligados à exportação podem, então, acessar esses cabeçalhos e propriedades do usuário. Consulte o tópico “Propagação” no centro de informações do WebSphere Integration Developer para obter informações adicionais.

Se esta for uma operação bidirecional, o componente retornará uma resposta.

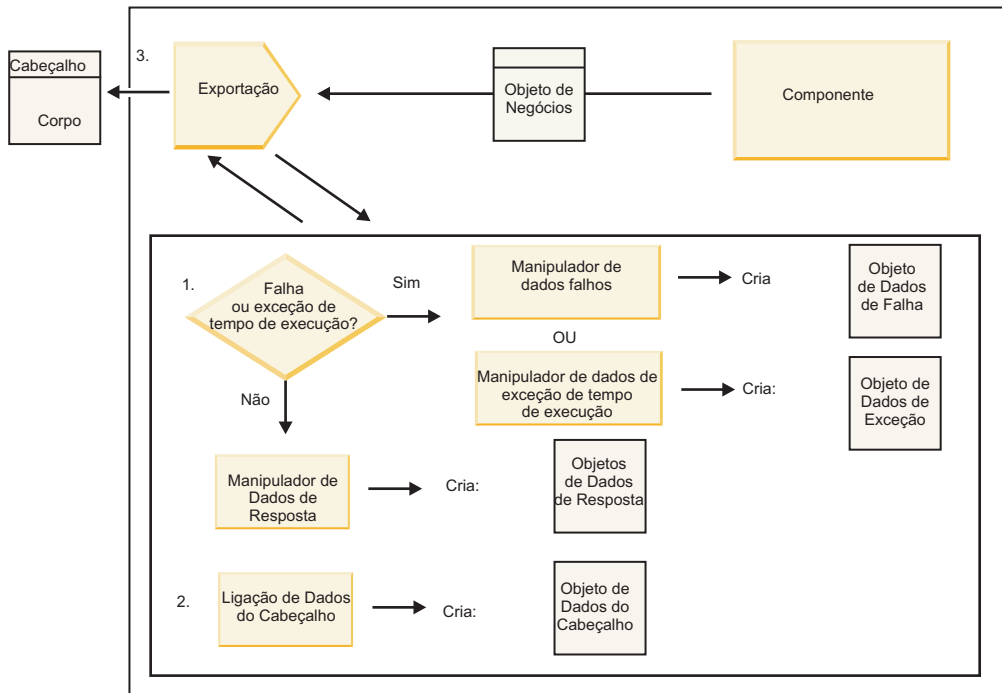


Figura 4. Fluxo de uma Resposta de Volta por meio da Exportação

A seguinte sequência de etapas ocorre:

1. Se uma mensagem de resposta normal é recebida pela ligação de exportação, o manipulador de dados de resposta ou a ligação de dados no método transforma o objeto de negócios em uma resposta. Se a resposta for uma falha, o manipulador de dados falhos ou a ligação de dados no método transformará a falha em uma resposta de falha. Para ligações de exportação HTTP apenas, se a resposta for uma exceção de tempo de execução, o manipulador de dados de exceção de tempo de execução, se configurado, será chamado.
2. Para ligações WebSphere MQ apenas, a ligação de dados de cabeçalho transforma os objetos de dados de cabeçalho em cabeçalhos de protocolo.
3. A exportação envia a resposta de serviço durante o transporte.

Fluxo de Informações por meio de uma Importação

Componentes enviam solicitações para serviços fora do módulo usando uma importação. A solicitação é enviada, por meio de um transporte específico determinado pela ligação associada.

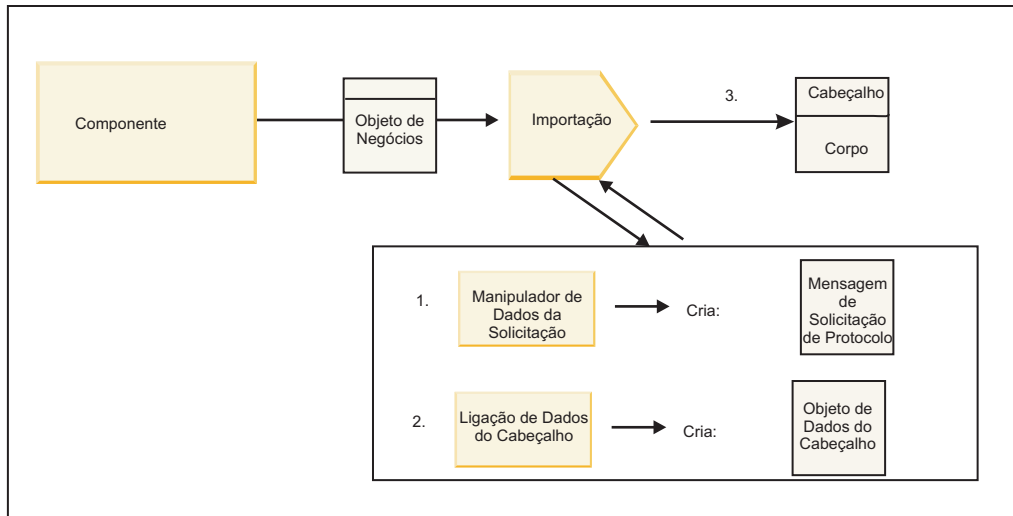


Figura 5. Fluxo a Partir de um Componente por meio da Importação para um Serviço

O componente chama a importação com um objeto de negócios de solicitação.

Nota:

- A ligação de importação HTTP, a ligação de importação do serviço da web e a ligação de importação EJB devem ser chamadas de forma síncrona chamando o componente.
- A ligação de importação JMS, JMS Genérica, JMS do MQ e do WebSphere MQ deve ser chamada assincronicamente.

Após o componente chamar a importação, a sequência de eventos a seguir ocorre:

1. O manipulador de dados da solicitação ou a ligação de dados no método transforma o objeto de negócios de solicitação em uma mensagem de solicitação de protocolo.
2. Para ligações do WebSphere MQ apenas, a ligação de dados do cabeçalho no método configurar o objeto de negócios do cabeçalho no cabeçalho do protocolo.
3. A importação chama o serviço com a solicitação de serviço durante o transporte.

Se esta for uma operação bidirecional, o serviço retornará uma resposta e a sequência de etapas a seguir ocorrerá:

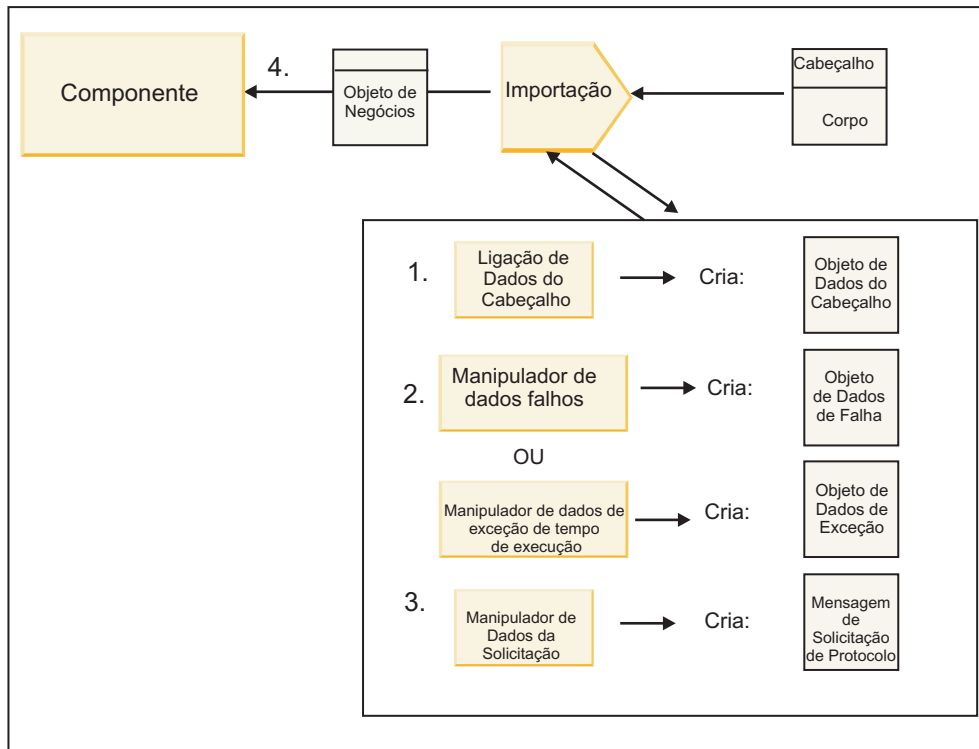


Figura 6. Fluxo de uma Resposta de Volta por meio da Importação

1. Para ligações do WebSphere MQ apenas, a ligação de dados do cabeçalho transforma o cabeçalho do protocolo em um objeto de dados do cabeçalho.
2. Uma determinação é feita sobre se a resposta é uma falha.
 - Se a resposta for uma falha, o seletor de falha inspecionará a falha para determinar para qual falha de WSDL ele mapeia. O manipulador de dados falhos no método, então, transforma a falha em uma resposta de falha.
 - Se a resposta for uma exceção de tempo de execução, o manipulador de dados de exceção de tempo de execução, se configurado, será chamado.
3. O manipulador de dados de resposta ou a ligação no método transforma a resposta em um objeto de negócios de resposta.
4. A importação retorna o objeto de negócios de resposta para o componente.

Configuração da Exportação e Importação de Ligação

Um dos aspectos principais das ligações de exportação e importação é a transformação do formato de dados, que indica como os dados são mapeados (desserializados) a partir de um formato de ligação nativo para um objeto de negócios ou como eles são mapeados (serializados) de um objeto de negócios para um formato de ligação nativo. Para ligações associadas às exportações, também é possível especificar um seletor de função para indicar qual operação deve ser executada nos dados. Para ligações associadas às exportações ou importações, é possível indicar como falhas que ocorrem durante o processamento devem ser manipuladas.

Além disso, especifique informações específicas de transporte nas ligações. Por exemplo, para uma ligação HTTP, especifique a URL do terminal. Para a ligação HTTP, as informações específicas de transporte são descritas nos tópicos “Gerando uma Ligação de Importação HTTP” e “Gerando uma Ligação de Exportação HTTP”. Também é possível localizar informações sobre outras ligações no centro de informações.

Transformação do Formato de Dados em Importações e Exportações:

Quando uma ligação de exportação ou importação é configurada no IBM Integration Designer, uma das propriedades de configuração que especificar é o formato de dados usado pela ligação.

- Para ligações de exportação, nas quais um aplicativo cliente envia solicitações e recebe respostas a partir de um componente SCA, você indica o formato dos dados nativos. Dependendo do formato, o sistema seleciona o manipulador de dados ou a ligação de dados apropriada para transformar os dados nativos em um objeto de negócios (que é usado pelo componente SCA) e, inversamente, para transformar o objeto de negócios em dados nativos (que é a resposta para o aplicativo cliente).
- Para ligações de importação, nas quais um componente SCA envia solicitações e recebe respostas de um serviço fora do módulo, você indica o formato de dados dos dados nativos. Dependendo do formato, o sistema seleciona o manipulador de dados ou a ligação de dados apropriada para transformar o objeto de negócios em dado nativo e vice-versa.

O IBM Business Process Manager fornece um conjunto de formatos de dados predefinidos e manipuladores de dados ou ligações de dados correspondentes que suportam os formatos. Também é possível criar seus próprios manipuladores de dados customizados e registrar o formato de dados para esses manipuladores de dados. Para obter informações adicionais, consulte o tópico “Desenvolvendo manipuladores de dados” no centro de informações do IBM Integration Designer.

- *Manipuladores de Dados* são de protocolo neutro e transformam dados de um formato para outro. No IBM Business Process Manager, os manipuladores de dados geralmente transformam dados nativos (tais como XML, CSV e COBOL) em um objeto de negócios e um objeto de negócios em dado nativo. Como eles são de protocolo neutro, é possível reutilizar o mesmo manipulador de dados com uma variedade de ligações de exportação e importação. Por exemplo, é possível usar o mesmo manipulador de dados XML com uma ligação de exportação ou importação HTTP ou com uma ligação de exportação ou importação JMS.
- *Ligações de Dados* também transformam dados nativos em um objeto de negócios (e vice-versa), mas elas são específicas de protocolo. Por exemplo, uma ligação de dados HTTP pode ser usada apenas com uma ligação de exportação ou de importação HTTP. Diferente de manipuladores de dados, uma ligação de dados HTTP não pode ser reutilizada com uma ligação de exportação ou importação do MQ.

Nota: Três ligações de dados HTTP (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML e HTTPServiceGatewayDataBinding) estão descontinuadas a partir do IBM Business Process Manager Versão 7.0. Use manipuladores de dados sempre que possível.

Conforme observado anteriormente, é possível criar manipuladores de dados customizados, se necessário. Também é possível criar ligações de dados customizadas; no entanto, é recomendado que você crie manipuladores de dados customizados porque eles podem ser usados em diversas ligações.

Manipuladores de Dados:

Os manipuladores de dados são configurados com relação às ligações de exportação e de importação para transformar dados de um formato para outro no estilo protocolo neutro. Vários manipuladores de dados são fornecidos como parte do produto, mas também é possível criar seu próprio manipulador de dados, se necessário. É possível associar um manipulador de dados a uma ligação de exportação ou de importação em um de dois níveis: é possível associá-lo à todas as operações na interface de exportação ou de importação ou é possível associá-lo a uma operação específica para solicitação ou resposta.

Manipuladores de Dados Predefinidos

Você usa o IBM Integration Designer para especificar o manipulador de dados que deseja usar.

Os manipuladores de dados que são predefinidos para seu uso são listados na seguinte tabela, que também descreve como cada manipulador de dados transforma os dados de entrada e de saída.

Nota: Exceto onde observado, esse manipuladores de dados podem ser usados com ligações JMS, JMS Genérico, JMS do MQ, WebSphere MQ e HTTP.

Consulte o tópico “Manipuladores de Dados” no centro de informações do Integration Designer para obter informações mais detalhadas.

Tabela 23. Manipuladores de Dados Predefinidos

Manipulador de dados	Dados nativos para objeto de negócios	Objeto de negócios para dados nativos
ATOM	Analisa os feeds ATOM em um objeto de negócios feed ATOM.	Serializa um objeto de negócios feed ATOM para os feeds ATOM.
Delimitado	Analisa dados delimitados em um objeto de negócios.	Serializa um objeto de negócios para dados delimitados, incluindo o CSV.
Largura Fixa	Analisa dados de largura fixa em um objeto de negócios.	Serializa um objeto de negócios para dados de largura fixa.
Manipulado pelo WTX	Delega transformação de formato de dados para o WebSphere Transformation Extender (WTX). O nome do mapa WTX é derivado pelo manipulador de dados.	Delega transformação de formato de dados para o WebSphere Transformation Extender (WTX). O nome do mapa WTX é derivado pelo manipulador de dados.
Manipulado pelo invocador de WTX	Delega transformação de formato de dados para o WebSphere Transformation Extender (WTX). O nome do mapa WTX é fornecido pelo usuário.	Delega transformação de formato de dados para o WebSphere Transformation Extender (WTX). O nome do mapa WTX é fornecido pelo usuário.
JAXB	Serializa os Java beans para um objeto de negócios usando as regras de mapeamento definidas pela especificação Java Architecture for XML Binding (JAXB).	Desserializa um objeto de negócios para os Java beans usando as regras de mapeamento definidas pela especificação JAXB.
JAXWS Nota: O manipulador de dados do JAXWS pode ser usado apenas com a ligação EJB.	Usado por uma ligação EJB para transformar um objeto Java de resposta ou um objeto Java de exceção para um objeto de negócios de resposta usando as regras de mapeamento definidas pela especificação Java API for XML Web Services (JAX-WS).	Usado por uma ligação EJB para transformar um objeto de negócios para os parâmetros do método Java de saída usando as regras de mapeamento definidas pela especificação JAX-WS.
JSON	Analisa dados JSON em um objeto de negócios.	Serializa um objeto de negócios para dados JSON.
Corpo nativo	Analisa os bytes, texto, mapa, fluxo ou objetos nativos em um dos cinco objetos de negócios de base (texto, bytes, mapa, fluxo ou objeto).	Transforma os cinco objetos de negócios de base em byte, texto, mapa, fluxo ou objeto.
SOAP	Analisa a mensagem SOAP (e o cabeçalho) em um objeto de negócios.	Serializa um objeto de negócios para uma mensagem SOAP.
XML	Analisa dados XML em um objeto de negócios.	Serializa um objeto de negócios para dados XML.
UTF8XMLDataHandler	Analisa dados XML codificados em UTF-8 em um objeto de negócios.	Serializa um objeto de negócios em dados XML codificados em UTF-8 quando enviar uma mensagem.

Criando um Manipulador de Dados

Informações detalhadas sobre a criação de um manipulador de dados podem ser localizadas no tópico “Desenvolvendo Manipuladores de Dados” no centro de informações do Integration Designer.

Ligações de Dados:

As ligações de dados são configuradas nas ligações de exportação e importação para transformar dados de um formato para outro. As ligações de dados são específicas de um protocolo. Várias ligações de dados são fornecidas como parte do produto, mas também é possível criar sua própria ligação de dados, se necessário. É possível associar uma ligação de dados a uma ligação de exportação ou importação em um de dois níveis - é possível associá-la a todas as operações na interface da exportação ou importação ou você pode associá-la a uma operação específica para a solicitação ou resposta.

É possível usar o IBM Integration Designer para especificar qual ligação de dados deseja usar ou criar sua própria ligação de dados. Uma discussão de como criar ligações de dados pode ser localizada na seção “Visão Geral das Ligações de JMS, MQ JMS e JMS Genérica” do centro de informações do IBM Integration Designer.

Ligações JMS

A tabela a seguir lista as ligações de dados que podem ser usadas com:

- Ligações JMS
- Ligações JMS Genéricas
- Ligações JMS do WebSphere MQ

A tabela também inclui uma descrição das tarefas que as ligações de dados executam.

Tabela 24. Ligações de Dados Predefinidas para Ligações JMS

Ligação de Dados	Dados nativos para objeto de negócios	Objeto de negócios para dados nativos
Objeto Java Serializado	Transforma o objeto serializado Java em um objeto de negócios (que é mapeado como o tipo de entrada ou saída no WSDL).	Serializa um objeto de negócios para o objeto serializado Java na mensagem do objeto JMS.
Bytes agrupados	Extrai os bytes da mensagem de bytes JMS recebida e os agrupa no objeto de negócios JMSBytesBody.	Extrai os bytes do objeto de negócios JMSBytesBody e os agrupa na mensagem de bytes JMS de saída.
Entrada de mapa agrupada	Extrai o nome, valor e as informações de tipo para cada entrada na mensagem do mapa JMS recebida e cria uma lista de objetos de negócios MapEntry. Em seguida, agrupa a lista no objeto de negócios JMSMapBody.	Extrai o nome, o valor e as informações de tipo da lista MapEntry no objeto de negócios JMSMapBody e cria as entradas correspondentes na mensagem do mapa JMS de saída.
Objeto agrupado	Extrai o objeto da mensagem do objeto JMS recebida e o agrupa no objeto de negócios JMSObjectBody.	Extrai o objeto do objeto de negócios JMSObjectBody e o agrupa na mensagem do objeto JMS de saída.
Texto agrupado	Extrai o texto da mensagem de texto de JMS recebida e o agrupa no objeto de negócios JMSTextBody.	Extrai o texto do objeto de negócios JMSTextBody e o agrupa na mensagem de texto de JMS de saída.

Ligações WebSphere MQ

A tabela a seguir lista as ligações de dados que podem ser usadas com o WebSphere MQ e descreve as tarefas que as ligações de dados executam.

Tabela 25. Ligações de Dados Predefinidas para Ligações do WebSphere MQ

Ligação de Dados	Dados nativos para objeto de negócios	Objeto de negócios para dados nativos
Objeto Java Serializado	Transforma o objeto serializado Java a partir da mensagem recebida em um objeto de negócios (que é mapeado como o tipo de entrada ou saída no WSDL).	Transforma um objeto de negócios no objeto serializado Java na mensagem de saída
Bytes agrupados	Extrai os bytes da mensagem de bytes do MQ não estruturada e os agrupa no objeto de negócios JMSBytesBody.	Extrai os bytes de um objeto de negócios JMSBytesBody e agrupa os bytes na mensagem de bytes do MQ não estruturada de saída.
Texto agrupado	Extrai o texto de uma mensagem de texto do MQ não estruturada e o agrupa em um objeto de negócios JMSTextBody.	Extrai texto a partir de um objeto de negócios JMSTextBody e o agrupa em uma mensagem de texto do MQ não estruturada.
Entrada do fluxo agrupada	Extrai as informações de nome e tipo para toda entrada na mensagem de fluxo do JMS recebida e cria uma lista dos objetos de negócios StreamEntry. Em seguida, agrupa a lista no objeto de negócios JMSStreamBody.	Extrai as informações de nome e tipo da lista StreamEntry no objeto de negócios JMSStreamBody e cria entradas correspondentes na JMSStreamMessage de saída.

Além das ligações de dados listadas em Tabela 25, o WebSphere MQ também usa ligações de dados de cabeçalho. Consulte o centro de informações do IBM Integration Designer para obter detalhes.

Ligações HTTP

A tabela a seguir lista as ligações de dados que podem ser usadas com HTTP e descreve as tarefas executadas pelas ligações de dados.

Tabela 26. Ligações de dados predefinidas para ligações HTTP

Ligação de Dados	Dados nativos para objeto de negócios	Objeto de negócios para dados nativos
Bytes agrupados	Extrai os bytes do corpo da mensagem HTTP recebida e os agrupa no objeto de negócios HTTPBytes.	Extrai os bytes do objeto de negócios HTTPBytes e os inclui no corpo da mensagem HTTP de saída.
Texto agrupado	Extrai o texto do corpo da mensagem HTTP recebida e o agrupa no objeto de negócios HTTPText.	Extrai o texto do objeto de negócios HTTPText e o inclui no corpo da mensagem HTTP de saída.

Seletores de Função em Ligações de Exportação:

Um seletor de função é usado para indicar qual operação deve ser executada nos dados para uma mensagem de solicitação. Os seletores de função são configurados como parte de uma ligação de exportação.

Considere uma exportação SCA que expõe uma interface. A interface contém duas operações - Criar e Atualizar. A exportação tem uma ligação JMS que lê a partir de uma fila.

Quando uma mensagem chega na fila, a exportação transmite os dados associados, mas qual operação da interface de exportação deve ser chamada no componente ligado? A operação é determinada pelo seletor de função e pela configuração de ligação de exportação.

O seletor de função retorna o nome da função nativa (o nome da função no sistema do cliente que enviou a mensagem). O nome da função nativa é, então, mapeado para o nome da operação ou função na interface associada à exportação. Por exemplo, na figura a seguir, o seletor de função retorna o nome da função nativa (CRT) a partir da mensagem recebida, o nome da função nativa é mapeado para a operação Criar e o objeto de negócios é enviado para o componente SCA com a operação Criar.

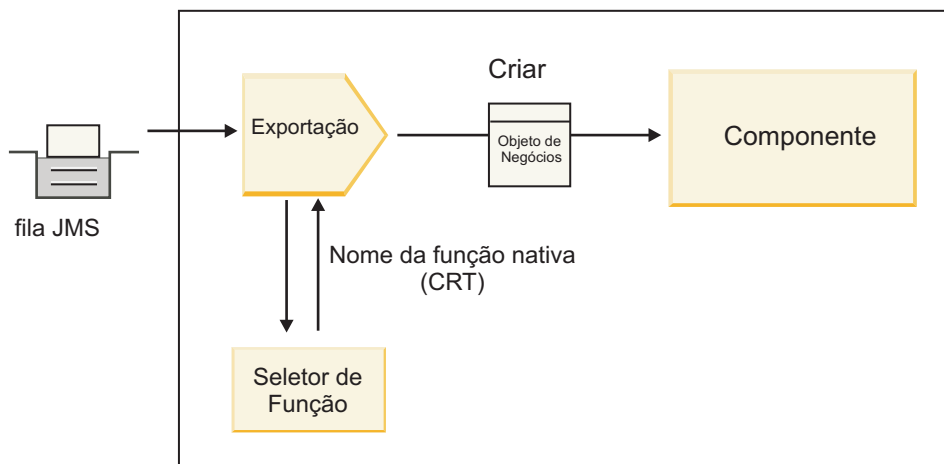


Figura 7. O Seletor de Função

Se a interface tiver apenas uma operação, não há necessidade de especificar um seletor de função.

Vários seletores de função pré-empacotados estão disponíveis e são listados nas seções a seguir.

Ligações JMS

A tabela a seguir lista os seletores de função que podem ser usados com:

- Ligações JMS
- Ligações JMS Genéricas
- Ligações JMS do WebSphere MQ

Tabela 27. Seletores de Função Predefinidos para Ligações JMS

Seletor de Função	Descrição
Seletor de Função JMS para Ligações de Dados do JMS Simples	Usa a propriedade JMSType da mensagem para selecionar o nome da operação.
Seletor de Função de Propriedade do Cabeçalho JMS	Retorna o valor da Propriedade de Sequência JMS, TargetFunctionName, do cabeçalho.
Seletor de Função do Gateway de Serviço JMS	Determina se a solicitação é uma operação unidirecional ou bidirecional examinando a propriedade JMSReplyTo configurada pelo cliente.

Ligações do WebSphere MQ

A tabela a seguir lista os seletores de função que podem ser usados com ligações do WebSphere MQ.

Tabela 28. Seletores de Função Predefinidos para Ligações do WebSphere MQ

Seletor de Função	Descrição
Seletor de Função handleMessage MQ	Retorna handleMessage como um valor, que é mapeado usando as ligações de método de exportação para o nome de uma operação na interface.
O MQ utiliza o seletor da função padrão JMS	Lê a operação nativa a partir da propriedade TargetFunctionName da pasta de um cabeçalho MQRFH2.
O MQ utiliza o formato do corpo da mensagem como função nativa	Localiza o campo Formato do último cabeçalho e retorna esse campo como uma Sequência.
Seletor de Função de Tipo MQ	Cria um método em sua ligação de exportação recuperando uma URL contendo as propriedades Msd, Set, Type e Format localizadas no cabeçalho MQRFH2.
Seletor de Função do Gateway de Serviço MQ	Usa a propriedade MsgType no cabeçalho MQMD para determinar o nome da operação.

Ligações HTTP

A tabela a seguir lista os seletores de função que podem ser usados com ligações HTTP.

Tabela 29. Seletores de Função Predefinidos para Ligações HTTP

Seletor de Função	Descrição
Seletor de Função HTTP com base em Cabeçalho TargetFunctionName	Usa a propriedade do cabeçalho de HTTP TargetFunctionName do cliente para determinar qual operação chamar no tempo de execução da exportação.
Seletor de Função HTTP Baseado na URL e no Método de HTTP	Usa o caminho relativo da URL anexada com o método de HTTP a partir do cliente para determinar a operação nativa definida na exportação.
Seletor de função de gateway de serviço HTTP baseado na URL com um nome da operação	Determina o método a ser chamado com base na URL se "operationMode = oneway" tiver sido anexado à URL de solicitação.

Nota: Também é possível criar seu próprio seletor de função, usando o IBM Integration Designer. Informações sobre como criar um seletor de função são fornecidas no centro de informações do IBM Integration Designer. Por exemplo, uma descrição de como criar um seletor de função para ligações do WebSphere MQ pode ser localizada em "Visão Geral dos Seletores de Função do MQ".

Manipulação de Falha:

É possível configurar suas ligações de importação e exportação para manipular falhas (por exemplo, exceções de negócios) que ocorrem durante o processamento especificando manipuladores de dados falhos. É possível configurar um manipulador de dados falhos em três níveis - você pode associar um manipulador de dados falhos a uma falha, a uma operação ou para todas as operações a uma ligação.

Um manipulador de dados falhos processa dados falhos e os transforma no formato correto para serem enviados pela ligação de exportação ou importação.

- Para uma ligação de exportação, o manipulador de dados falhos transforma o objeto de negócios de exceção enviado do componente para uma mensagem de resposta que pode ser usada pelo aplicativo cliente.

- Para uma ligação de importação, o manipulador de dados falhos transforma os dados falhos ou a mensagem de resposta enviada de um serviço para um objeto de negócios de exceção que pode ser usado pelo componente SCA.

Para ligações de importação, a ligação chama o seletor de falha, que determina se a mensagem de resposta é uma resposta normal, uma falha de negócios ou uma exceção de tempo de execução.

É possível especificar um manipulador de dados falhos para uma determinada falha, para uma operação e para todas as operações com uma ligação.

- Se o manipulador de dados falhos estiver configurado em todos os três níveis, o manipulador de dados associado a uma determinada falha será chamado.
- Se os manipuladores de dados falhos estiverem configurados nos níveis da operação e da ligação, o manipulador de dados associado à operação será chamado.

Dois editores são usados no IBM Integration Designer para especificar a manipulação de falhas. O editor de interface é usado para indicar se haverá uma falha em uma operação. Depois que uma ligação é gerada com esta interface, o editor na visualização de propriedades permite configurar como o falha será manipulada. Para obter informações adicionais, consulte o tópico “Seletores de Falha” no centro de informações do IBM Integration Designer.

Como as Falhas São Manipuladas em Ligações de Exportação:

Quando ocorre uma falha durante o processamento da solicitação de um aplicativo cliente, a ligação de exportação pode retornar as informações de falha para o cliente. Configure a ligação de exportação para especificar como a falha deve ser processada e retornada ao cliente.

Configure a ligação de exportação usando IBM Integration Designer.

Durante o processamento da solicitação, um cliente chama uma exportação com uma solicitação e a exportação chama o componente SCA. Durante o processamento da solicitação, o componente SCA pode retornar uma resposta de negócios ou pode emitir uma exceção de negócios de serviço ou uma exceção de tempo de execução de serviço. Quando isso ocorre, a ligação de exportação transforma a exceção em uma mensagem de falha e a envia para o cliente, conforme mostrado na figura a seguir e descrito nas seções a seguir.

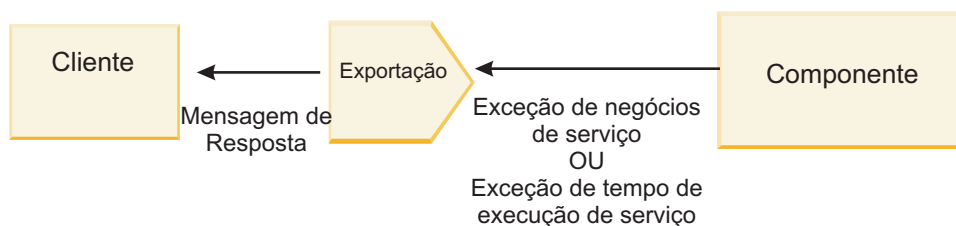


Figura 8. Como Informações de Falha são Enviadas a Partir do Componente Através da Ligação de Exportação para o Cliente

É possível criar um manipulador de dados ou uma ligação de dados customizada para manipular falhas.

Falhas de Negócios

Falhas de negócios são erros de negócios ou exceções que ocorrem durante o processamento.

Considere a interface a seguir, que possui uma operação createCustomer nela. Esta operação possui duas falhas de negócios definidas: CustomerAlreadyExists e MissingCustomerId.

Operações

Operações e seus parâmetros

	Nome	Tipo
createCustomer		
Entrada(s)	entrada	CustomerInfo
Saída(s)	saída	CustomerInfo
Falha	Cliente Já Existe	Customer Already ExistsBO
Falha	MissingCustomerID	MissingCustomerIDBO

Figura 9. Interface com Duas Falhas

Neste exemplo, se um cliente enviar uma solicitação para criar um cliente (para este componente SCA) e se o cliente já existir, o componente lançará uma falha `CustomerAlreadyExists` para a exportação. A exportação precisa propagar esta falha de negócios de volta para o cliente de chamada. Para fazer isso, ela usa o manipulador de dados falhos que está configurado na ligação de exportação.

Quando uma falha de negócios é recebida pela ligação de exportação, ocorre o seguinte processamento:

1. A ligação determina qual manipulador de dados falhos chamar para manipular a falha. Se a exceção de negócios de serviço contém o nome da falha, o manipulador de dados que está configurado na falha é chamado. Se a exceção de negócios de serviço não contém o nome da falha, o nome da falha é derivado correspondendo os tipos de falha.
2. A ligação chama o manipulador de dados falhos com o objeto de dados da exceção de negócios de serviço.
3. O manipulador de dados falhos transforma o objeto de dados falhos em uma mensagem de resposta e a retorna para a ligação de exportação.
4. A exportação retorna a mensagem de resposta para o cliente.

Se a exceção de negócios de serviço contém o nome da falha, o manipulador de dados que está configurado na falha é chamado. Se a exceção de negócios de serviço não contém o nome da falha, o nome da falha é derivado correspondendo os tipos de falha.

Exceções de Tempo de Execução

Uma exceção de tempo de execução é uma exceção que ocorre no aplicativo SCA durante o processamento de uma solicitação que não corresponde a uma falha de negócios. Ao contrário de falhas de negócios, exceções de tempo de execução não são definidas na interface.

Em determinados cenários, você pode desejar propagar essas exceções de tempo de execução para o aplicativo cliente para que o aplicativo cliente possa executar a ação apropriada.

Por exemplo, se um cliente envia uma solicitação (para o componente SCA) para criar um cliente e um erro de autorização ocorre durante o processamento da solicitação, o componente emite uma exceção de tempo de execução. Esta exceção de tempo de execução precisa ser propagada de volta ao cliente de chamada para que ele possa executar a ação apropriada com relação à autorização. Isso é realizado pelo manipulador de dados de exceção de tempo de execução configurado na ligação de exportação.

Nota: É possível configurar um manipulador de dados de exceção de tempo de execução somente em ligações HTTP.

O processamento de uma exceção de tempo de execução é semelhante ao processamento de uma falha de negócios. Se um manipulador de dados de exceção de tempo de execução foi configurado, o processamento a seguir ocorrerá:

1. A ligação de exportação chama o manipulador de dados apropriado com a exceção de tempo de execução de serviço.
2. O manipulador de dados transforma o objeto de dados falhos em uma mensagem de resposta e a retorna para a ligação de exportação.
3. A exportação retorna a mensagem de resposta para o cliente.

A manipulação de falha e a manipulação de exceção de tempo de execução são opcionais. Se você não desejar propagar falhas ou exceções de tempo de execução para o cliente de chamada, não configure o manipulador de dados falhos ou o manipulador de dados de exceção de tempo de execução.

Como as Falhas São Manipuladas em Ligações de Importação:

Um componente usa uma importação para enviar uma solicitação para um serviço fora do módulo. Quando ocorre uma falha durante o processamento da solicitação, o serviço retorna a falha para a ligação de importação. É possível configurar a ligação de importação para especificar como a falha deve ser processada e retornada ao componente.

Configure a ligação de importação usando o IBM Integration Designer. É possível especificar um manipulador de dados falhos (ou ligação de dados) e você também especifica um seletor de falha.

Manipuladores de Dados Falhos

O serviço que processa a solicitação envia, para a ligação de importação, informações de falha na forma de uma exceção ou uma mensagem de resposta que contém os dados da falha.

A ligação de importação transforma a exceção de serviço ou mensagem de resposta em uma exceção de negócios do serviço ou exceção de tempo de execução de serviço, conforme mostrado na figura a seguir e descrito nas seções a seguir.

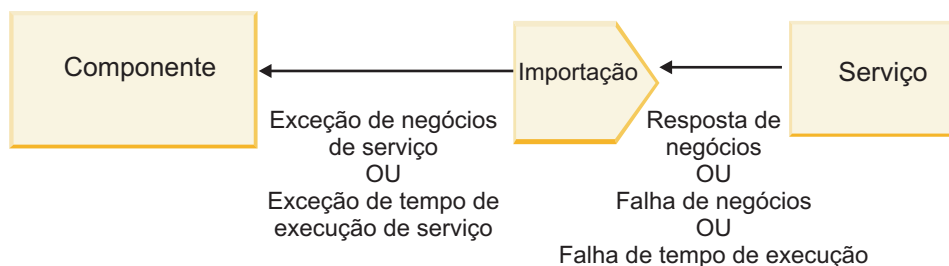


Figura 10. Como as Informações de Falha são Enviadas a Partir do Serviço Através da Importação para o Componente

É possível criar um manipulador de dados ou uma ligação de dados customizada para manipular falhas.

Seletores de Falha

Quando você configura uma ligação de importação, pode especificar um seletor de falha. O seletor de falha determina se a resposta de importação é uma resposta real, uma exceção de negócios ou uma falha de tempo de execução. Ele também determina, a partir do corpo de resposta ou cabeçalho, o nome da falha nativa, que é mapeado pela configuração de ligação para o nome de uma falha na interface associada.

Dois tipos de seletores de falha pré-empacotados estão disponíveis para uso com as importações de JMS, JMS do MQ, JMS Genéricas, do WebSphere MQ e HTTP:

Tabela 30. Seletores de Falha Pré-empacotados

Tipo de Seletor de Falha	Descrição
Baseado em Cabeçalho	Determina se uma mensagem de resposta é uma falha de negócios, uma exceção de tempo de execução ou uma mensagem normal baseada nos cabeçalhos na mensagem de resposta recebida.
SOAP	Determina se a mensagem SOAP de resposta é uma resposta normal, falha de negócios ou exceção de tempo de execução.

A seguir são mostrados exemplos de seletores de falha baseados em cabeçalho e do seletor de falha de SOAP.

- Seletor de Falha Baseado em Cabeçalho

Se um aplicativo deseja indicar que a mensagem de entrada é uma falha de negócios, deve haver dois cabeçalhos na mensagem de entrada para falhas de negócios, o que é mostrado como segue:

```
Header name = FaultType, Header value = Business
Header name = FaultName, Header value = <user defined native fault name>
```

Se um aplicativo desejar indicar que a mensagem de resposta recebida é uma exceção de tempo de execução, deverá haver um cabeçalho na mensagem recebida, o que é mostrado como segue:

```
Header name = FaultType, Header value = Runtime
```

- Seletor de falhas de SOAP

Uma falha de negócios pode ser enviada como parte da mensagem SOAP com o seguinte cabeçalho SOAP customizado. "CustomerAlreadyExists" é o nome da falha neste caso.

```
<ibmSoap:BusinessFaultName
xmlns:ibmSoap="http://www.ibm.com/soap">CustomerAlreadyExists
</ibmSoap:BusinessFaultName>
```

O seletor de falha é opcional. Se você não especificar um seletor de falha, a ligação de importação não poderá determinar o tipo de resposta. A ligação, portanto, a tratará como uma resposta de negócios e chamará o manipulador de dados de resposta ou ligação de dados.

É possível criar um seletor de falha customizado. As etapas para criar um seletor de falha customizado são fornecidas no tópico "Desenvolvendo um Seletor de Falha Customizado" do centro de informações do IBM Integration Designer.

Falhas de Negócios

Uma falha de negócios pode ocorrer quando há um erro no processamento de uma solicitação. Por exemplo, se você enviar uma solicitação para criar um cliente e esse cliente já existir, o serviço enviará uma exceção de negócios para a ligação de importação.

Quando uma exceção de negócios é recebida pela ligação, as etapas de processamento dependem de se um seletor de falha foi configurado para a ligação.

- Se nenhum seletor de falha foi configurado, a ligação chamará a ligação de dados ou manipulador de dados de resposta.
- Se um seletor de falha foi configurado, ocorrerá o seguinte processamento:
 1. A ligação de importação chama o seletor de falha para determinar se a resposta é falha de negócios, resposta de negócios ou falha no tempo de execução.
 2. Se a resposta for uma falha de negócios, a ligação de importação chamará o seletor de falha para fornecer o nome de falha nativo.
 3. A ligação de importação determina a falha WSDL correspondente ao nome de falha nativo retornado pelo seletor de falha.

4. A ligação de importação determina o manipulador de dados falhos que está configurado para esta falha de WSDL.
5. A ligação de importação chama este manipulador de dados falhos com os dados falhos.
6. O manipulador de dados falhos transforma os dados falhos em um objeto de dados e o retorna para a ligação de importação.
7. A ligação de importação constrói um objeto de exceção de negócios de serviço com o objeto de dados e o nome da falha.
8. A importação retorna o objeto de exceção de negócios de serviço para o componente.

Exceções de Tempo de Execução

Uma exceção de tempo de execução pode ocorrer quando há um problema na comunicação com o serviço. O processamento de uma exceção de tempo de execução é semelhante ao processamento de uma exceção de negócios. Se um seletor de falha foi configurado, ocorrerá o seguinte processamento:

1. A ligação de importação chama o manipulador de dados de exceção de tempo de execução apropriado com os dados da exceção.
2. O manipulador de dados de exceção de tempo de execução transforma os dados de exceção em um objeto de exceção de tempo de execução de serviço e o retorna para a ligação de importação.
3. A importação retorna o objeto de exceção de tempo de execução de serviço para o componente.

Interoperabilidade entre módulos SCA e serviços SCA abertos

O IBM WebSphere Application Server v7.0 Feature Pack para Service Component Architecture (SCA) fornece um modelo de programação simples, porém poderoso, para construir aplicativos com base nas especificações de SCA Aberto. Os módulos SCA do IBM Business Process Manager usam ligações de importação e exportação para interoperar com serviços de SCA Aberto desenvolvidos em um ambiente do Rational Application Developer e hospedados pelo WebSphere Application Server Feature Pack para Service Component Architecture.

Um aplicativo SCA chama um aplicativo SCA Aberto por meio de uma ligação de importação. Um aplicativo SCA recebe uma chamada de um aplicativo SCA Aberto por meio de uma ligação de exportação. Uma lista de ligações suportadas é mostrada no “Chamando Serviços sobre Ligações Interoperáveis” na página 71.

Chamando os Serviços de SCA Aberto a Partir de Módulos SCA

Aplicativos SCA desenvolvidos com o IBM Integration Designer podem chamar aplicativos SCA Abertos desenvolvidos em um ambiente do Rational Application Developer. Esta seção fornece um exemplo de como chamar um serviço SCA Aberto a partir de um módulo SCA usando uma ligação de importação SCA.

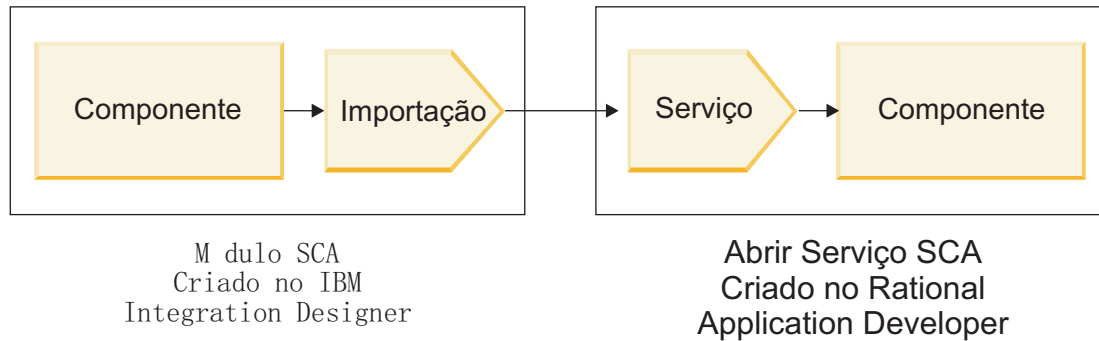


Figura 11. Componente no Módulo SCA Chamando Serviço SCA Aberto

Nenhuma configuração especial é necessária para chamar um serviço SCA Aberto.

Para se conectar a um serviço SCA Aberto por meio de uma ligação de importação SCA, você fornece o nome do componente e nome do serviço SCA Aberto na ligação de importação.

1. Para obter o nome do componente de destino e serviço do SCA Aberto composto, execute as seguintes etapas:
 - a. Assegure que a guia **Propriedades** esteja aberta clicando em **Janela > Mostrar Visualização > Propriedades**.
 - b. Abra o editor composto dando um clique duplo no diagrama composto que contém o componente e o serviço. Por exemplo, para um componente denominado **customer**, o diagrama composto é **customer.composite_diagram**.
 - c. Clique no componente de destino.
 - d. No campo **Nome** da guia **Propriedades**, anote o nome do componente de destino.
 - e. Clique no ícone de serviço associado ao componente.
 - f. No campo **Nome** da guia **Propriedades**, anote o nome do serviço.
2. Para configurar a importação do IBM Business Process Manager para conectá-lo ao serviço SCA Aberto, execute as seguintes etapas:
 - a. No IBM Integration Designer, navegue até a guia **Propriedades** da importação SCA que você deseja conectar ao serviço SCA Aberto.
 - b. No campo **Nome do Módulo**, insira o nome do componente a partir da etapa 1d.
 - c. No campo **Nome de Exportação**, insira o nome do serviço a partir da etapa 1f.
 - d. Salve seu trabalho pressionando Ctrl+S.

Chamando Módulos SCA a Partir de Serviços SCA Abertos

Aplicativos SCA Abertos desenvolvidos em um ambiente do Rational Application Developer podem chamar aplicativos SCA desenvolvidos com o IBM Integration Designer. Esta seção fornece um exemplo de como chamar um módulo SCA (por meio de uma ligação de exportação SCA) a partir de um serviço SCA Aberto.

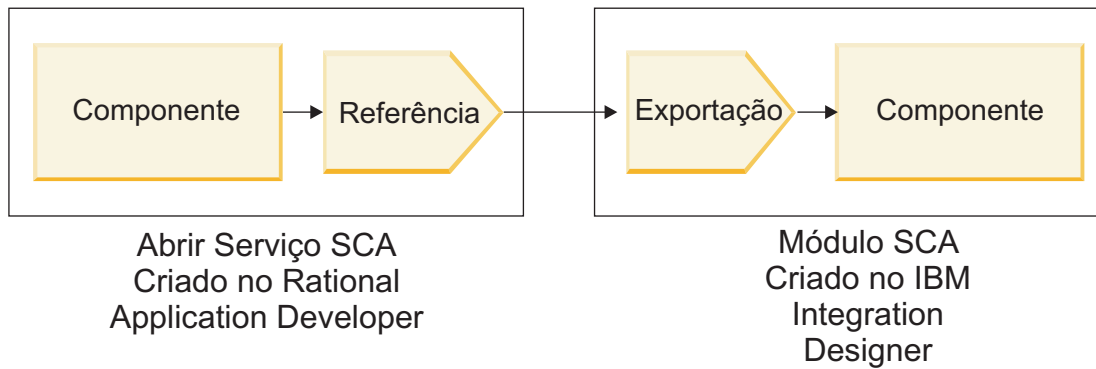


Figura 12. Serviço SCA Aberto Chamando Componente no Módulo SCA

Para se conectar a um componente SCA por meio de uma ligação de referência de SCA Aberto, você fornece o nome do módulo e o nome de exportação.

1. Para obter o nome do módulo de destino e da exportação, execute as seguintes etapas:
 - a. No IBM Integration Designer, abra o módulo no editor de montagem dando um clique duplo no módulo.
 - b. Clique na exportação.
 - c. No campo **Nome** da guia **Propriedades**, anote o nome da exportação.
2. Configure a referência de SCA Aberto que você deseja conectar ao módulo e à exportação do IBM Business Process Manager:
 - a. No Rational Application Developer, abra o editor composto dando um clique duplo no diagrama composto que contém o componente e o serviço.
 - b. Clique no ícone de referência da referência de componente para exibir as propriedades de referência na guia **Propriedades**.
 - c. Clique na guia **Ligação** no lado esquerdo da página.
 - d. Clique em **Ligações** e, em seguida, clique em **Incluir**.
 - e. Selecione a ligação **SCA**.
 - f. No campo **URI**, insira o nome do módulo do IBM Business Process Manager, seguido de uma barra ("/"), seguido pelo nome de exportação (que você determinou na etapa 1c).
 - g. Clique em **OK**.
 - h. Salve seu trabalho pressionando Ctrl+S.

Chamando Serviços sobre Ligações Interoperáveis

As seguintes ligações são suportadas para interoperabilidade com um serviço SCA Aberto.

- **Ligação SCA**
 No IBM Business Process Manager, quando um módulo SCA chama um serviço SCA Aberto por meio de uma ligação de importação SCA, os seguintes estilos de chamada são suportados:
 - Assíncrono (unidirecional)
 - Síncrono (solicitação/resposta)
 A interface de importação SCA e a interface de serviço Open SCA devem usar uma interface WSDL compatível com interoperabilidade de serviços da web (WS-I).
 Observe que a ligação SCA suporta propagação de transação e contexto de segurança.
- **Ligação de serviço da Web (JAX-WS) com o protocolo SOAP1.1/HTTP ou SOAP1.2/HTTP**
 A interface de importação SCA e a interface de serviço Open SCA devem usar uma interface WSDL compatível com interoperabilidade de serviços da web (WS-I).
 Além disso, as seguintes qualidades de serviço são suportadas:

- Transação Atômica de Serviços da Web
- Segurança de Serviços da Web
- Ligação EJB

Uma interface Java é usada para definir a interação entre um módulo SCA e um serviço SCA Aberto quando a ligação EJB é usada.

Observe que a ligação EJB suporta propagação de transação e contexto de segurança.
- Ligações JMS

A interface de importação SCA e a interface de serviço Open SCA devem usar uma interface WSDL compatível com interoperabilidade de serviços da web (WS-I).

Os seguintes provedores JMS são suportados:

 - WebSphere Platform Messaging (Ligação JMS)
 - WebSphere MQ (Ligação JMS do MQ)

Nota: Gráficos de Negócios não são interoperáveis em nenhuma ligação SCA e, portanto, não são suportados em interfaces usadas para interoperar com o WebSphere Application Server Feature Pack para Service Component Architecture.

Tipos de Ligação

As *ligações* específicas do protocolo são usadas com importações e exportações para especificar o meio de transportar dados dentro ou fora de um módulo.

Selecionando Ligações Apropriadas:

Ao criar um aplicativo, é necessário saber como selecionar a ligação mais apropriada às necessidades do seu aplicativo.

As ligações disponíveis no IBM Integration Designer fornecem uma série de opções. Nessa lista, é possível determinar qual tipo de ligação pode ser a mais adequada às necessidades do seu aplicativo.

Considere uma ligação *Service Component Architecture (SCA)* quando esses fatores são aplicáveis:

- Todos os serviços estão contidos em módulos, ou seja, não há nenhum serviço externo.
- Você deseja separar a função em diferentes módulos SCA que interagem diretamente entre si.
- Os módulos estão firmemente acoplados.

Considere uma ligação de *serviço da web* quando estes fatores são aplicáveis:

- É necessário acessar um serviço externo sobre a Internet ou fornecer um serviço sobre a Internet.
- Os serviços estão fracamente acoplados.
- É preferível a comunicação síncrona, isto é, um pedido de um serviço pode aguardar pela resposta de outro.
- O protocolo do serviço externo que você está acessando ou o serviço que você deseja fornecer é SOAP/HTTP ou SOAP/JMS.

Considere uma ligação *HTTP* quando esses fatores forem aplicáveis:

- É necessário acessar um serviço externo sobre a Internet ou fornecer um serviço sobre a Internet e você está trabalhando com outros serviços da web como GET, PUT e DELETE.
- Os serviços estão fracamente acoplados.
- É preferível a comunicação síncrona, isto é, um pedido de um serviço pode aguardar pela resposta de outro.

Considere uma ligação *Enterprise JavaBeans (EJB)* quando esses fatores são aplicáveis:

- A ligação é para um serviço importado que é por si só um EJB ou que precisa ser acessado por clientes EJB.
- O serviço importado está fracamente acoplado.
- As interações EJB stateful não são necessárias.
- É preferível a comunicação síncrona, isto é, um pedido de um serviço pode aguardar pela resposta de outro.

Considere uma ligação *Enterprise Information Systems (EIS)* quando esses fatores são aplicáveis:

- É necessário acessar um serviço em um sistema EIS usando um adaptador de recursos.
- A transmissão de dados síncrona é preferencial sobre a assíncrona.

Considere uma ligação *Java Message Service (JMS)* quando esses fatores são aplicáveis:

Importante: Existem vários tipos de ligações JMS. Se espera trocar mensagens SOAP usando JMS, considere a ligação do serviço da web com o protocolo SOAP/JMS. Consulte o “Ligações de Serviço da Web” na página 74.

- É necessário acessar um sistema de mensagens.
- Os serviços estão fracamente acoplados.
- A transmissão de dados assíncrona é preferencial sobre a síncrona.

Considere uma ligação *Generic Java Message Service (JMS)* quando esses fatores são aplicáveis:

- É necessário acessar um sistema de mensagens do fornecedor não IBM.
- Os serviços estão fracamente acoplados.
- A confiabilidade é mais importante do que o desempenho; ou seja, a transmissão de dados assíncrona é preferida sobre a síncrona.

Considere uma ligação *Message Queue (MQ)* quando esses fatores são aplicáveis:

- É necessário acessar um sistema de mensagens do WebSphere MQ e é necessário usar as funções nativas do MQ.
- Os serviços estão fracamente acoplados.
- A confiabilidade é mais importante do que o desempenho; ou seja, a transmissão de dados assíncrona é preferida sobre a síncrona.

Considere uma ligação *MQ JMS* quando esses fatores forem aplicáveis:

- É necessário acessar um sistema de mensagens do WebSphere MQ, mas pode fazê-lo em um contexto de JMS, isto é, o subconjunto de funções do JMS é suficiente para seu aplicativo.
- Os serviços estão fracamente acoplados.
- A confiabilidade é mais importante do que o desempenho; ou seja, a transmissão de dados assíncrona é preferida sobre a síncrona.

Ligações SCA:

Uma ligação de Service Component Architecture (SCA) permite que um serviço se comunique com outros serviços em outros módulos. Uma importação com uma ligação SCA permite acessar um serviço em outro módulo SCA. Uma exportação com uma ligação SCA permite oferecer um serviço para outros módulos.

Você pode usar o IBM Integration Designer para gerar e configurar ligações do SCA em importações e exportações nos módulos do SCA.

Se os módulos estiverem em execução no mesmo servidor ou forem implementados no mesmo cluster, uma ligação de SCA será a ligação mais fácil e mais rápida a usar.

Depois que o módulo que contém a ligação SCA é implementado no servidor, é possível usar o console administrativo para visualizar informações sobre a ligação ou, no caso de uma ligação de importação, para alterar propriedades selecionadas da ligação.

Ligações de Serviço da Web:

Uma ligação de serviços da web é o meio de transmitir mensagens de um componente Service Component Architecture (SCA) para um serviço da web (e vice-versa).

Visão geral das ligações de serviço da Web:

Uma ligação de importação do serviço da web permite chamar um serviço da web externo de seus componentes Service Component Architecture (SCA). Uma ligação de exportação de serviço da web permite expor seus componentes SCA a clientes como serviços da web.

Com uma ligação de serviço da web, você acessa serviços externos usando mensagens SOAP e qualidades de serviço (QoS) interoperáveis.

Você usa o Integration Designer para gerar e configurar ligações de serviço da web em importações e exportações em módulos SCA. Os tipos de ligações de serviços da web a seguir estão disponíveis:

- SOAP1.2/HTTP e SOAP1.1/HTTP

Essas ligações são baseadas em Java API for XML Web Services (JAX-WS), uma API de programação Java para criar serviços da web.

- Use o SOAP1.2/HTTP se seu serviço da web estiver em conformidade com a especificação SOAP 1.2.
- Use o SOAP1.1/HTTP se seu serviço da web estiver em conformidade com a especificação SOAP 1.1.

Importante: Ao implementar um aplicativo com uma ligação de serviço da web (JAX-WS), o servidor de destino não deve ter a opção **Iniciar componentes conforme necessário** selecionada. Acesse “Verificando a Configuração do Servidor” na página 83 para obter detalhes.

Quando você seleciona uma destas ligações, é possível enviar anexos com suas mensagens SOAP.

As ligações de serviço da web funcionam com mensagens SOAP padrão. Usando uma das ligações JAX-WS de serviço da web, no entanto, é possível customizar a forma como as mensagens SOAP são analisadas ou gravadas. Por exemplo, é possível manipular elementos não padrão em mensagens SOAP ou aplicar o processamento adicional na mensagem SOAP. Quando você configura a ligação, especifica um manipulador de dados customizado que executa esse processamento na mensagem SOAP.

É possível usar conjuntos de políticas com uma ligação de serviço da web (JAX-WS). Um conjunto de políticas é uma coleção de tipos de política, cada um dos quais fornece uma qualidade de serviço. Por exemplo, o conjunto de políticas WSAddressing fornece uma forma de transporte neutro para endereçar uniformemente serviços da web e mensagens. Você pode usar o Integration Designer para selecionar o conjunto de políticas para a ligação.

Nota: Se desejar usar um conjunto de políticas Security Assertion Markup Language (SAML), você deverá executar alguma configuração adicional, conforme descrito em “Importando Conjuntos de Políticas SAML” na página 80.

- SOAP1.1/HTTP

Use esta ligação se desejar criar serviços da web que usam uma mensagem codificada por SOAP baseada em Java API for XML-based RPC (JAX-RPC).

- SOAP1.1/JMS

Use esta ligação para enviar ou receber mensagens SOAP usando um destino de Sistema de Mensagens Java (JMS).

Independente do transporte (HTTP ou JMS) que é usado para transmitir a mensagem SOAP, as ligações de serviço da web sempre manipulam interações de solicitação/resposta de forma síncrona. O encadeamento que faz a chamada no provedor de serviços é bloqueado até que uma resposta seja recebida do provedor. Consulte “Chamada Síncrona” para obter informações adicionais sobre esse estilo de chamada.

Importante: As seguintes combinações de ligações de serviço da web não podem ser usadas nas exportações no mesmo módulo. Se precisar expor componentes usando mais de uma destas ligações de exportação, você precisará ter cada um em um módulo separado e, em seguida, conectar esses módulos em seus componentes usando a ligação SCA:

- SOAP 1.1/JMS e SOAP 1.1/HTTP usando JAX-RPC
- SOAP 1.1/HTTP usando JAX-RPC e SOAP 1.1/HTTP usando JAX-WS
- SOAP 1.1/HTTP usando JAX-RPC e SOAP 1.2/HTTP usando JAX-WS

Após o módulo SCA que contém a ligação de serviço da web ser implementado no servidor, é possível usar o console administrativo para visualizar informações sobre a ligação ou alterar as propriedades selecionadas da ligação.

Nota: Os serviços da Web permitem que aplicativos interoperem usando descrições padrão de serviços e formatos padrão para as mensagens que eles trocam. Por exemplo, as ligações de importação e exportação de serviço da web podem interoperar com serviços que são implementados usando Web Services Enhancements (WSE) Versão 3.5 e Windows Communication Foundation (WCF) Versão 3.5 para Microsoft .NET. Ao interoperar com tais serviços, você deve assegurar que:

- O arquivo Web Services Description Language (WSDL) que é usado para acessar uma exportação de serviço da web inclui um valor de ação SOAP não vazio para cada operação na interface.
- O cliente de serviço da web configura o cabeçalho SOAPAction ou o cabeçalho wsa:Action ao enviar mensagens a uma exportação de serviço da web.

Propagação de Cabeçalho SOAP:

Ao manipular mensagens SOAP, pode ser necessário acessar informações de determinados cabeçalhos SOAP em mensagens que são recebidas, assegurar que as mensagens com cabeçalhos SOAP sejam enviadas com valores específicos ou permitir que os cabeçalhos SOAP passem por meio de um módulo.

Ao configurar uma ligação de serviço da web no Integration Designer, é possível indicar que deseja que os cabeçalhos SOAP sejam propagados.

- Quando as solicitações são recebidas em uma exportação ou respostas são recebidas em uma importação, as informações do cabeçalho SOAP podem ser acessadas, permitindo que a lógica no módulo seja baseada em valores de cabeçalho e permitindo que esses cabeçalhos sejam modificados.
- Quando solicitações são enviadas de uma exportação ou respostas são enviadas de uma importação, cabeçalhos SOAP podem ser incluídos nessas mensagens.

O formato e a presença dos cabeçalhos SOAP propagados podem ser afetados pelos conjuntos de política configurados na importação ou exportação, conforme explicado em Tabela 31 na página 77.

Para configurar a propagação de cabeçalhos SOAP para uma importação ou exportação, você seleciona (a partir da visualização Propriedades do Integration Designer) a guia **Propagar Cabeçalho do Protocolo** e seleciona as opções requeridas.

Cabeçalho WS-Addressing

O cabeçalho WS-Addressing pode ser propagado pela ligação de serviço da web (JAX-WS).

Quando você propagar o cabeçalho WS-Addressing, esteja ciente das seguintes informações:

- Se você ativar a propagação para o cabeçalho WS-Addressing, o cabeçalho será propagado no módulo nas seguintes circunstâncias:
 - Quando as solicitações são recebidas em uma exportação
 - Quando as respostas são recebidas em uma importação
- O cabeçalho WS-Addressing não é propagado para mensagens de saída a partir do IBM Business Process Manager (ou seja, o cabeçalho não é propagado quando as solicitações são enviadas a partir de uma importação ou quando as respostas são enviadas a partir da exportação).

Cabeçalho WS-Security

O cabeçalho WS-Security pode ser propagado pela ligação de serviço da web (JAX-WS) e a ligação de serviço da web (JAX-RPC).

A especificação WS-Security de serviços da web descreve aprimoramentos no sistema de mensagens SOAP para fornecer uma qualidade de proteção por meio da integridade da mensagem, da confidencialidade da mensagem e da autenticação de mensagem única. Esses mecanismos podem ser usados para acomodar uma grande variedade de modelos de segurança e tecnologias de criptografia.

Ao propagar o cabeçalho WS-Security, observe as seguintes informações:

- Se você ativar a propagação para o cabeçalho WS-Security, o cabeçalho será propagado por meio do módulo nas seguintes circunstâncias:
 - Quando as solicitações são recebidas em uma exportação
 - Quando solicitações são enviadas a partir de uma importação
 - Quando as respostas são recebidas em uma importação
- O cabeçalho *não* será, por padrão, propagado quando as respostas forem enviadas a partir da exportação. No entanto, se você configurar a propriedade JVM **WSSECURITY.ECHO.ENABLED** como **true**, o cabeçalho será propagado quando as respostas forem enviadas a partir da exportação. Nesse caso, se o cabeçalho WS-Security no caminho da solicitação não for modificado, os cabeçalhos WS-Security poderão ser ecoados automaticamente das solicitações para as respostas.
- O formato exato da mensagem SOAP enviada a partir de uma importação para uma solicitação ou a partir de uma exportação para uma resposta poderá não corresponder exatamente à mensagem SOAP que foi recebida originalmente. Por esse motivo, qualquer assinatura digital deve ser assumida para se tornar inválida. Se uma assinatura digital for requerida em mensagens que são enviadas, ela deverá ser estabelecida usando o conjunto de políticas de segurança apropriado e cabeçalhos WS-Security relacionados à assinatura digital em mensagens recebidas devem ser removidos dentro do módulo.

Para propagar o cabeçalho WS-Security, você deve incluir o esquema WS-Security com o módulo aplicativo. Consulte “Incluindo o Esquema WS-Security em um Módulo Aplicativo” na página 77 para conhecer o procedimento para incluir o esquema.

Como os Cabeçalhos são Propagados

A forma com que os cabeçalhos são propagados depende da configuração da política de segurança na ligação de importação ou exportação, conforme mostrado no Tabela 31 na página 77:

Tabela 31. Como Cabeçalhos de Segurança são Transmitidos

	Ligação de exportação sem política de segurança	Ligação de exportação com política de segurança
Ligação de importação sem política de segurança	<p>Os cabeçalhos de segurança são transmitidos no estado em que se encontram através do módulo. Eles não são descriptografados.</p> <p>Os cabeçalhos são enviados no mesmo formato em que foram recebidos.</p> <p>A assinatura digital pode se tornar inválida.</p>	<p>Os cabeçalhos de segurança são descriptografados e passados pelo módulo com verificação de assinatura e autenticação.</p> <p>Os cabeçalhos descriptografados são enviados.</p> <p>A assinatura digital pode se tornar inválida.</p>
Ligação de importação com política de segurança	<p>Os cabeçalhos de segurança são transmitidos no estado em que se encontram através do módulo. Eles não são descriptografados.</p> <p>Os cabeçalhos não devem ser propagados para a importação. Caso contrário, ocorre um erro devido à duplicação.</p>	<p>Os cabeçalhos de segurança são descriptografados e passados pelo módulo com verificação de assinatura e autenticação.</p> <p>Os cabeçalhos não devem ser propagados para a importação. Caso contrário, ocorre um erro devido à duplicação.</p>

Configure os conjuntos de políticas apropriados nas ligações de exportação e importação, pois isso isola o solicitante de serviços das alterações na configuração ou requisitos de QoS do provedor de serviços. Tendo cabeçalhos SOAP padrão visíveis em um módulo, eles podem, então, ser usados para influenciar o processamento (por exemplo, criação de log e rastreamento) no módulo. Propagar os cabeçalhos SOAP em um módulo a partir de uma mensagem recebida para uma mensagem enviada significa que os benefícios de isolamento do módulo são reduzidos.

Cabeçalhos padrão, como cabeçalhos WS-Security, não devem ser propagados em uma solicitação para uma importação ou resposta para uma exportação quando a importação ou exportação possui um conjunto de política associado que normalmente resultaria na geração desses cabeçalhos. Caso contrário, ocorrerá um erro devido a uma duplicação dos cabeçalhos. Em vez disso, os cabeçalhos devem ser explicitamente removidos ou a ligação de importação ou exportação deve ser configurada para evitar a propagação dos cabeçalhos de protocolo.

Acessando Cabeçalhos SOAP

Quando uma mensagem que contém os cabeçalhos SOAP é recebida de uma importação ou exportação de serviço da web, os cabeçalhos são posicionados na seção do cabeçalho do objeto de mensagem de serviço (SMO). É possível acessar as informações de cabeçalho, conforme descrito em "Acessando Informações de Cabeçalho SOAP no SMO".

Incluindo o Esquema WS-Security em um Módulo Aplicativo

O procedimento a seguir descreve as etapas para incluir o esquema no módulo aplicativo:

- Se o computador no qual o Integration Designer está em execução tem acesso à Internet, execute as seguintes etapas:
 1. Na perspectiva de Integração de Negócios, selecione **Dependências** para seu projeto.
 2. Expanda **Recursos Predefinidos** e selecione **Arquivos de Esquema WS-Security 1.0** ou **Arquivos de Esquema WS-Security 1.1** para importar o esquema em seu módulo.
 3. Limpe e reconstrua o projeto.

- Se um computador no qual o Integration Designer está executando não tiver acesso à Internet, você poderá fazer download do esquema para um segundo computador que tenha acesso à Internet. É possível, então, copiá-lo no computador no qual o Integration Designer está em execução.

1. No computador que possui acesso à Internet, faça download do esquema remoto:
 - a. Clique em **Arquivo > Importar > Integração de Negócios > WSDL e XSD**.
 - b. Selecione **WSDL Remoto** ou **Arquivo XSD**.
 - c. Importe os esquemas a seguir:
 - `http://www.w3.org/2003/05/soap-envelope/`
 - `http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd`
 - `http://www.w3.org/TR/xmlsig-core/xmlsig-core-schema.xsd`
2. Copie os esquemas no computador que não possui acesso à Internet.
3. No computador que não possui acesso à Internet, importe o esquema:
 - a. Clique em **Arquivo > Importar > Integração de Negócios > WSDL e XSD**.
 - b. Selecione **WSDL Local** ou **Arquivo XSD**.
4. Altere os locais de esquema para `oasis-wss-wssecurity-secext-1.1.xsd`:
 - a. Abra o esquema em `workplace_location/module_name/StandardImportFilesGen/oasis-wss-wssecurity-secext-1.1.xsd`.
 - b. Altere:


```
<xs:import namespace='http://www.w3.org/2003/05/soap-envelope'
  schemaLocation='http://www.w3.org/2003/05/soap-envelope'/>
```

 para:


```
<xs:import namespace='http://www.w3.org/2003/05/soap-envelope'
  schemaLocation='../w3/_2003/_05/soap_envelope.xsd'/>
```
 - c. Altere:


```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
  schemaLocation='http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd'/>
```

 para:


```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
  schemaLocation='../w3/tr/_2002/rec_xmlenc_core_20021210/xenc-schema.xsd'/>
```
5. Altere o local de esquema para `oasis-200401-wss-wssecurity-secext-1.0.xsd`:
 - a. Abra o esquema em `workplace_location/module_name/standardimportfilesgen/oasis-200401-wss-wssecurity-secext-1.0.xsd`.
 - b. Altere:


```
<xsd:import namespace="http://www.w3.org/2000/09/xmlsig#"
  schemaLocation="http://www.w3.org/TR/xmlsig-core/xmlsig-core-schema.xsd"/>
```

 para:


```
<xsd:import namespace="http://www.w3.org/2000/09/xmlsig#"
  schemaLocation="../w3/tr/_2002/rec_xmlsig_core_20020212/xmlsig-core-schema.xsd"/>
```
6. Limpe e reconstrua o projeto.

Propagação do Cabeçalho de Transporte:

Ao manipular mensagens SOAP, pode ser necessário acessar informações de determinados cabeçalhos de transporte em mensagens que são recebidas, assegurar que as mensagens com cabeçalhos transport sejam enviadas com valores específicos ou permitir que os cabeçalhos de transporte passem por meio de um módulo.

Ao configurar uma ligação de serviço da web no Integration Designer, é possível indicar que deseja que os cabeçalhos de transporte sejam propagados.

- Quando solicitações são recebidas em uma exportação ou respostas são recebidas em uma importação, as informações do cabeçalho de transporte podem ser acessadas, permitindo que a lógica no módulo seja baseada nos valores de cabeçalho e permitindo que esses cabeçalhos sejam modificados.
- Quando respostas são enviadas a partir de uma exportação ou solicitações são enviadas a partir de uma importação, os cabeçalhos de transporte podem ser incluídos nessas mensagens.

Especificando a Propagação de Cabeçalhos

Para configurar a propagação de cabeçalhos de transporte para uma importação ou exportação, execute as seguintes etapas:

1. Na visualização Propriedades do Integration Designer, selecione **Ligação > Propagação**.
2. Configure a opção de propagação de cabeçalho de transporte que você requer.

Nota: A propagação do cabeçalho de transporte é desativada por padrão e pode ser implementada apenas em um ambiente de tempo de execução Versão 7.0.0.3 (ou posterior). Além disso, observe que, para a Versão 7.0.0.3, a propagação de cabeçalho de transporte é limitada a cabeçalhos de transporte HTTP apenas.

Se você ativar a propagação de cabeçalhos de transporte, os cabeçalhos serão propagados em um módulo a partir de mensagens recebidas e, se você não remover explicitamente os cabeçalhos, eles serão usados nas chamadas subsequentes no mesmo encadeamento.

Nota: Os cabeçalhos de transporte não podem ser propagados ao usar a ligação de serviço da web (JAX-RPC).

Acessando as Informações de Cabeçalho

Quando a propagação de cabeçalho de transporte é ativada para mensagens recebidas, todos os cabeçalhos de transporte (incluindo cabeçalhos definidos pelo cliente) ficam visíveis no objeto de mensagem do serviço (SMO). É possível configurar os cabeçalhos com valores diferentes ou criar novos. Observe, no entanto, que não há verificação ou validação dos valores configurados e qualquer cabeçalho impróprio ou incorreto pode causar problemas de tempo de execução de serviço da web.

Considere as seguintes informações sobre a configuração de cabeçalhos HTTP:

- Quaisquer mudanças nos cabeçalhos reservados para o mecanismo de serviço da web não serão aceitas na mensagem de saída. Por exemplo, a versão de HTTP ou método, os cabeçalhos Content-Type, Content-Length e SOAPAction serão reservados para o mecanismo do serviço da web.
- Se o valor do cabeçalho for um número, o número (em vez da sequência) deverá ser configurado diretamente. Por exemplo, use **Max-Forwards = 5** (em vez de **Max-Forwards = Max-Forwards: 5**) e **Age = 300** (em vez de **Age = Age: 300**).
- Se a mensagem de solicitação for menor que 32 KB de tamanho, o mecanismo de serviço da web removerá o cabeçalho Transfer-Encoding e no lugar configurará o cabeçalho Content-Length para o tamanho fixo da mensagem.
- O Content-language é reconfigurado por was.channel.http no caminho de resposta.
- Uma configuração inválida para resultados de Upgrade em um erro 500.
- Os cabeçalhos a seguir anexam o valor reservado pelo mecanismo de serviço da web para as configurações do cliente:
 - User-Agent
 - Cache-Control
 - Pragma
 - Accept
 - Conexão

É possível acessar as informações do cabeçalho de uma das maneiras a seguir:

- Usando uma primitiva de mediação para acessar as estruturas de SMO
Consulte os links “Informações Relacionadas” para localizar informações sobre como usar primitivas de mediação.
- Usando o serviço de contexto SPI

O código de amostra a seguir lê os cabeçalhos de transporte HTTP do serviço de contexto:

```
HeadersType headerType = ContextService.INSTANCE.getHeaders();
HTTPHeaderType httpHeaderType = headerType.getHTTPHeader();
List HTTPHeader httpHeaders = httpHeaderType.getHeader();
if(httpHeaders!=null){
    for(HTTPHeader httpHeader: httpHeaders){
        String httpHeadername = httpHeader.getName();
        String httpHeaderValue = httpHeader.getValue();
    }
}
List PropertyType properties = headerType.getProperties();
if(properties!=null){
    for(PropertyType property: properties){
        String propertyName = property.getName();
        String propertyValue = property.getValue().toString();
    }
}
```

Resolução de problemas

Se você encontrar problemas ao enviar os cabeçalhos revisados, poderá interceptar a mensagem TCP/IP usando ferramentas tais como o Monitor TCP/IP no Integration Designer. Você acessa o Monitor TCP/IP selecionando **Executar/Depurar > Monitor TCP/IP** na página Preferências.

Também é possível visualizar os valores de cabeçalho usando o rastreamento do mecanismo do JAX-WS:
org.apache.axis2.*=all: com.ibm.ws.websvcs.*=all:

Trabalhando com Ligações de Serviço da Web (JAX-WS):

Ao usar as ligações de serviço da web (JAX-WS) com seus aplicativos, é possível incluir uma qualidade de serviço (QOS) Security Assertion Markup Language (SAML) na ligação. Você deve primeiro usar o console administrativo para importar o conjunto de políticas. Também é possível usar o console administrativo para certificar-se de que o servidor esteja corretamente configurado para uso com a ligação de serviço da web (JAX-WS).

Importando Conjuntos de Políticas SAML:

Security Assertion Markup Language (SAML) é um padrão OASIS baseado em XML para trocar informações de identidade do usuário e de atributos de segurança. Quando configurar uma ligação de serviço da web (JAX-WS) no Integration Designer, é possível especificar um conjunto de políticas SAML. Primeiro use o console administrativo do IBM Business Process Manager para tornar os conjuntos de políticas SAML disponíveis para que eles possam ser importados no Integration Designer.

Os conjuntos de políticas SAML estão geralmente localizados no diretório de configuração do perfil:

profile_root/config/templates/PolicySets

Antes de iniciar este procedimento, verifique se os seguintes diretórios (que contêm os conjuntos de políticas) estão localizados no diretório de configuração do perfil:

- SAML11 Bearer WSHTTPS padrão
- SAML20 Bearer WSHTTPS padrão
- SAML11 Bearer WSSecurity padrão

- SAML20 Bearer WSSecurity padrão
- SAML11 HoK Public WSSecurity padrão
- SAML20 HoK Public WSSecurity padrão
- SAML11 HoK Symmetric WSSecurity padrão
- SAML20 HoK Symmetric WSSecurity padrão
- Padrão WSHTTPS de Nome do Usuário

Se os diretórios não estiverem no diretório de configuração do perfil, copie-os nesse diretório a partir do seguinte local:

`app_server_root/profileTemplates/default/documents/config/templates/PolicySets`

Você pode importar os conjuntos de política no console administrativo, selecionar aqueles que deseja disponibilizar no Integration Designer e salvar um arquivo .zip para cada um dos conjuntos de política em um local que seja acessível ao Integration Designer.

1. Importe os conjuntos de políticas seguindo estas etapas:
 - a. No console administrativo, clique em **Serviços > Conjuntos de Políticas > Conjuntos de Políticas de Aplicativo**.
 - b. Clique em **Importar > A Partir do Repositório Padrão**.
 - c. Selecione os conjuntos de políticas padrão SAML e clique em **OK**.
2. Exporte os conjuntos de políticas para que possam ser usados pelo Integration Designer:
 - a. Na página Conjuntos de Políticas de Aplicativo, selecione o conjunto de políticas SAML que deseja exportar e clique em **Exportar**.

Nota: Se a página Conjuntos de Políticas de Aplicativo não for exibida atualmente, clique em **Serviços > Conjuntos de Políticas > Conjuntos de Políticas de Aplicativo** a partir do console administrativo.

- b. Na próxima página, clique no link do arquivo .zip para o conjunto de políticas.
- c. Na janela Download de Arquivo, clique em **Salvar** e indique um local que esteja acessível para o Integration Designer.
- d. Clique em **Voltar**.
- e. Conclua as etapas 2a a 2d para cada conjunto de políticas que deseja exportar.

Os conjuntos de políticas SAML são salvos nos arquivos .zip e estão prontos para serem importados no Integration Designer.

Importe os conjuntos de políticas no Integration Designer, conforme descrito no tópico “Conjuntos de Políticas”.

Chamando Serviços da Web que Requerem Autenticação HTTP Básica:

A autenticação básica de HTTP utiliza um nome de usuário e uma senha para autenticar um cliente de serviço em um terminal seguro. É possível configurar a autenticação básica de HTTP ao enviar ou receber solicitações de serviços da Web.

Você pode configurar a autenticação básica de HTTP para receber solicitações de serviço da web configurando a ligação de exportação da API Java para Serviços da Web XML (JAX-WS), conforme descrito em Criando e designando funções de segurança a exportações de serviço da web.

Autenticação básica de HTTP pode ser ativada para as solicitações de serviço da Web que são enviadas por uma ligação de importação JAX-WS em uma de duas maneiras:

- Ao configurar a ligação de importação em um módulo SCA, é possível selecionar o conjunto de políticas de autenticação HTTP fornecido denominado BPMHTTPBasicAuthentication (que é fornecido com a ligação de importação de serviço da Web (JAX-WS)) ou qualquer outro conjunto de políticas que inclua a política HTTPTransport.
- Ao construir o módulo SCA, é possível usar os recursos de fluxo de mediação para criar dinamicamente um novo cabeçalho de autenticação HTTP e especificar as informações de nome de usuário e senha no cabeçalho.

Nota: O conjunto de políticas tem precedência sobre o valor especificado no cabeçalho. Se você deseja usar o valor configurado no cabeçalho de autenticação HTTP no tempo de execução, não conecte um conjunto de políticas que inclua a política HTTPTransport. Especificamente, não use o conjunto de políticas BPMHTTPBasicAuthentication padrão e, se tiver definido um conjunto de políticas, certifique-se de que ele exclua a política HTTPTransport.

Para obter mais informações sobre os conjuntos de políticas de serviço da web e as ligações de diretrizes e como são usados, consulte Conjuntos de políticas de serviços da web do Centro de Informações do WebSphere Application Server.

- Para usar o conjunto de políticas fornecido, execute as etapas a seguir:
 1. Opcional: No console administrativo, crie uma ligação de diretrizes gerais do cliente ou edite uma existente que inclua a política HTTPTransport com os valores de ID de usuário e senha necessários.
 2. No IBM Integration Designer, gere uma ligação de importação de serviço da Web (JAX-WS) e anexe o conjunto de políticas BPMHTTPBasicAuthentication.
 3. Execute *uma* das etapas a seguir:
 - No IBM Integration Designer, nas propriedades de ligação de importação de serviço da Web (JAX-WS), especifique o nome de uma ligação de política geral de cliente existente que inclua a política HTTPTransport.
 - Após implementar o módulo SCA, use o console administrativo para selecionar uma ligação de diretrizes do cliente existente ou crie uma nova ligação de diretrizes do cliente e associe-a à ligação de importação.
 4. Opcional: No console administrativo do servidor de processo, edite a ligação do conjunto de políticas selecionada para especificar o ID e a senha necessários.
- Para especificar o nome de usuário e a senha no cabeçalho de autenticação HTTP, execute um dos seguintes conjuntos de etapas:
 - Use a primitiva de mediação do Configurador de Cabeçalho HTTP no IBM Integration Designer para criar o cabeçalho de autenticação HTTP e especifique o nome do usuário e a senha.
 - Se for necessária lógica adicional, use o código Java em uma primitiva de mediação customizada (conforme mostrado no exemplo a seguir) para:
 1. Criar um cabeçalho de autenticação HTTP.
 2. Especificar as informações de nome de usuário e senha.
 3. Incluir o novo cabeçalho de autenticação HTTP em HTTPControl.
 4. Configurar o HTTPControl atualizado de volta no serviço de Contexto.

```
//Get the HeaderInfoType from contextService
ContextService contextService = (ContextService) ServiceManager.INSTANCE
.locateService("com/ibm/bpm/context/ContextService");
HeaderInfoType headers = contextService.getHeaderInfo();
if(headers == null){
    headers = ContextObjectFactory.eINSTANCE.createHeaderInfoType();
}
//Get the HTTP header and HTTP Control from HeaderInfoType
HTTPHeaderType httpHeaderType = headers.getHTTPHeader();
HTTPControl cp = httpHeaderType.getControl();
HeadersFactory factory = HeadersFactory.eINSTANCE;
if(cp == null){
    cp = factory.createHTTPControl();
}
```

```

}
//Create new HTTPAuthentication and set the HTTPCredentials
HTTPAuthentication authorization = factory.createHTTPAuthentication();
HTTPCredentials credentials = factory.createHTTPCredentials();
authorization.setAuthenticationType(HTTPAuthenticationType.BASIC_LITERAL);
credentials.setUserId("USERNAME");
credentials.setPassword("PASSWORD");
authorization.setCredentials(credentials);
cp.setAuthentication(authorization);
httpHeaderType.setControl(cp);
// Set header info back to the current execution context.
contextService.setHeaderInfo(headers);

```

Verificando a Configuração do Servidor:

Ao implementar um aplicativo com uma ligação de serviço da web (JAX-WS), você deve certificar-se de que o servidor no qual o aplicativo está implementado não tenha a opção **Iniciar componentes como necessário** selecionada.

É possível verificar se esta opção está selecionada executando as seguintes etapas a partir do console administrativo:

1. Clique em **Servidores > Tipos de Servidor > WebSphere Application Servers**.
2. Clique no nome do servidor.
3. Na guia Configuração, determine se **Iniciar Componentes Conforme Necessário** está selecionado.
4. Execute uma das etapas a seguir:
 - Se **Iniciar Componentes Conforme Necessário** estiver selecionado, remova a seleção e, em seguida, clique em **Aplicar**.
 - Se **Iniciar Componentes Conforme Necessário** não estiver selecionado, clique em **Cancelar**.

Anexos em mensagens SOAP:

É possível enviar e receber mensagens SOAP que incluem dados binários (como arquivos PDF ou imagens JPEG) como anexos. Anexos podem ser *referenciados* (ou seja, representados como partes da mensagem explicitamente na interface de serviço) ou *não referenciados* (em que números e tipos arbitrários de anexos podem ser incluídos).

Um anexo referenciado pode ser representado de uma das maneiras a seguir:

- Anexos MTOM usam a codificação especificada do SOAP Message Transmission Optimization Mechanism (<http://www.w3.org/TR/soap12-mtom/>). Anexos MTOM são ativados por meio de uma opção de configuração nas ligações de importação e exportação e são a maneira recomendada para codificar anexos para novos aplicativos.
- Como um elemento wsi:swaRef-typed no esquema de mensagem
Anexos definidos usando o tipo wsi:swaRef em conformidade com o Web Services Interoperability Organization (WS-I) *Attachments Profile Versão 1.0* (<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), que define como elementos de mensagem são relacionados a partes MIME.
- Como uma parte da mensagem de nível superior, usando um tipo de esquema binário
Anexos representados como partes da mensagem de nível superior em conformidade com a especificação *Mensagens SOAP com Anexos* (<http://www.w3.org/TR/SOAP-attachments>).
Anexos representados como partes da mensagem de nível superior também podem ser configurados para assegurar que o documento WSDL e as mensagens produzidos pela ligação estejam em conformidade com o *WS-I Attachments Profile Versão 1.0* e o *WS-I Basic Profile Versão 1.1* (<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>).

Um anexo não referenciado é transportado em uma mensagem SOAP sem qualquer representação no esquema de mensagem.

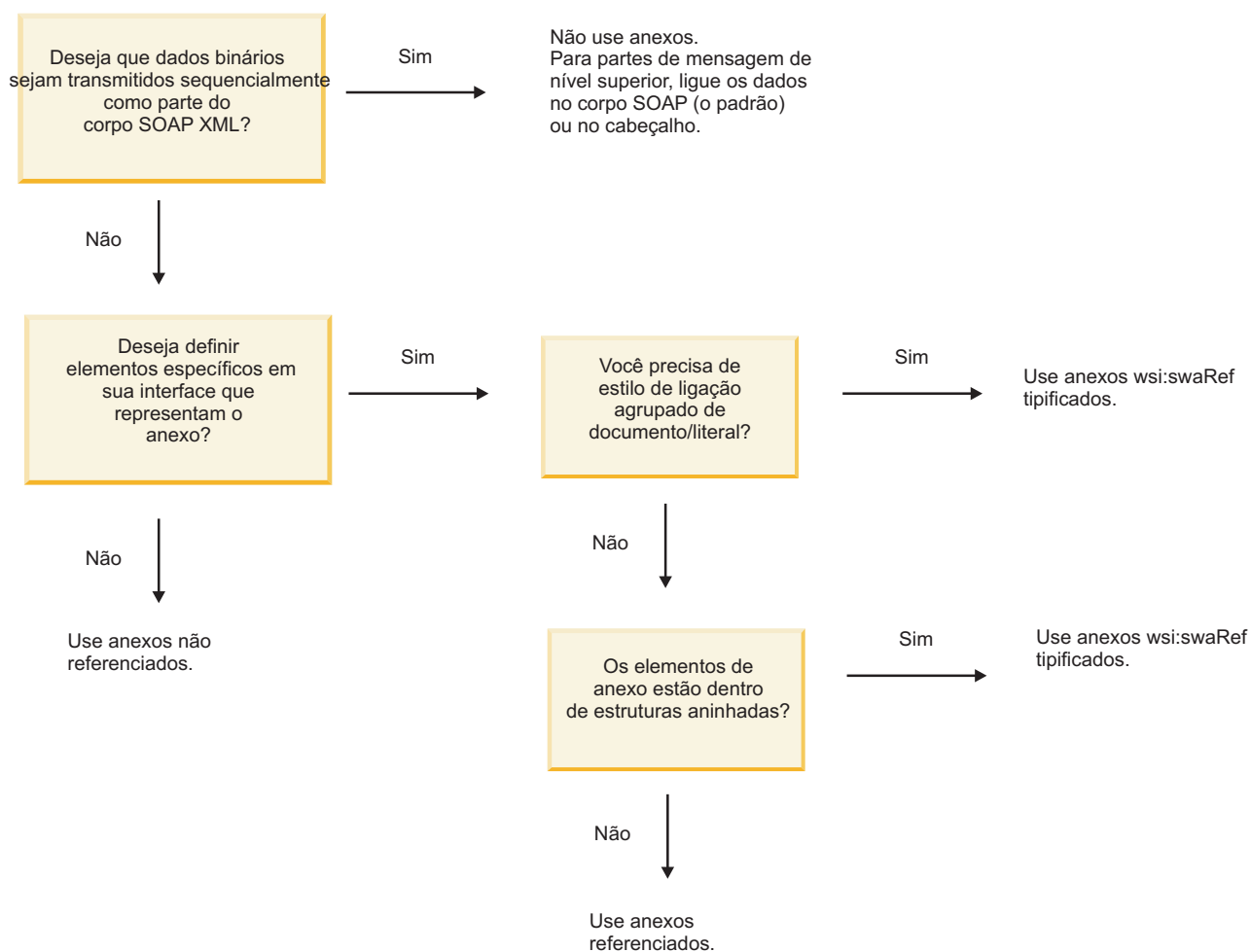
Em todos os casos, exceto anexos MTOM, a ligação WSDL SOAP deve incluir uma ligação MIME para anexos a serem usados, e o tamanho máximo dos anexos não deve exceder 20 MB.

Nota: Para enviar ou receber mensagens SOAP com anexos, você deve usar uma das ligações de serviços da web baseadas em Java API for XML Web Services (JAX-WS).

Como Escolher o Estilo Adequado de Anexo:

Ao projetar uma nova interface de serviço que inclui dados binários, considere como esses dados binários são transportados nas mensagens SOAP que são enviadas e recebidas pelo serviço.

O Message Transmission Optimization Mechanism (MTOM) deve ser usado para anexos se o aplicativo de serviços da web conectá-lo suportá-lo. Se não, o seguinte diagrama mostra como outros estilos de anexo são escolhidos. Use o seguinte conjunto de perguntas para determinar o estilo de anexo apropriado:



Anexos MTOM: Partes da Mensagem em Nível Superior:

É possível enviar e receber mensagens de serviço da web que incluem anexos SOAP Message Transmission Optimization Mechanism (MTOM). Em uma mensagem SOAP MIME com múltiplas partes, o corpo SOAP é a primeira parte da mensagem e o anexo ou anexos estão em partes subsequentes.

Ao enviar ou receber um anexo referenciado em uma mensagem SOAP, os dados binários que compõem o anexo (que é geralmente muito grande) são mantidos separadamente do corpo da mensagem SOAP

para que não precisem ser analisados como XML. Isso resulta em processamento mais eficiente do que se os dados binários fossem mantidos dentro de um elemento XML.

A seguir há uma amostra de uma mensagem SOAP MTOM:

```
... outros cabeçalhos de transporte ...
Content-Type: multipart/related; boundary=MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812;
type="application/xop+xml"; start="
<0.urn:uuid:0FE43E4D025F0BF3DC11582467646813@apache.org>";
start-info="text/xml"; charset=UTF-8

--MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812
content-type: application/xop+xml; charset=UTF-8; type="text/xml";
content-transfer-encoding: binary
content-id:
  <0.urn:uuid:0FE43E4D025F0BF3DC11582467646813@apache.org>

<?xml version="1.0" encoding="UTF-8"?>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header/>
    <soapenv:Body>
      <sendImage xmlns="http://org.apache.axis2/jaxws/sample/mtom">
        <input>
          <imageData><xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
href="cid:1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org"/></imageData>
        </input>
      </sendImage>
    </soapenv:Body>
  </soapenv:Envelope>
--MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812
content-type: text/plain
content-transfer-encoding: binary
content-id:
  <1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org>

... dados binários vão aqui ...
--MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812--
```

Note que na amostra de MTOM, o tipo de conteúdo para o envelope SOAP é **application/xop+xml** e os dados binários são substituídos por um elemento **xop:Include**, conforme abaixo:

```
<xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include" href="cid:1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org"/>
```

Processamento de Entrada de Anexos Referenciados

Quando um cliente passa uma mensagem SOAP com um anexo a um componente Service Component Architecture (SCA), a ligação de exportação (JAX-WS) de serviço da web primeiro remove o anexo. Ela, então, analisa a parte SOAP da mensagem e cria um objeto de negócios. Finalmente, a ligação configura o binário de anexo no objeto de negócios.

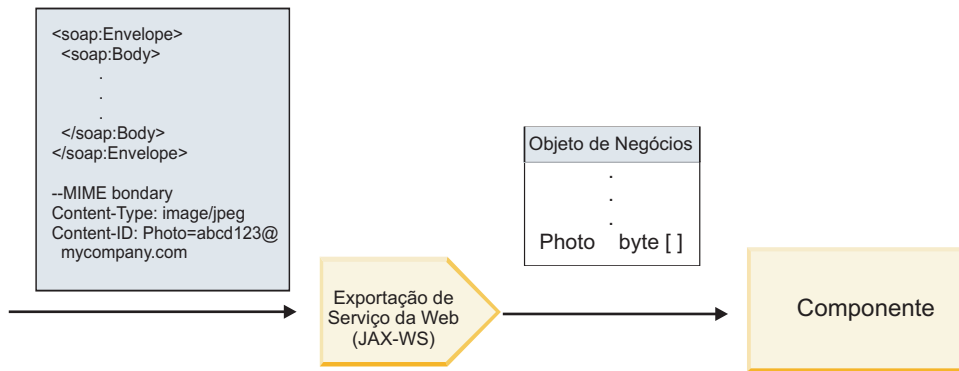


Figura 13. Como a Ligação de Exportação (JAX-WS) de Serviço da Web Processa uma Mensagem SOAP com um Anexo Referenciado

Atributos do Anexo MTOM

- O MTOM pode suportar elementos de anexo dentro de estruturas aninhadas.
- O MTOM somente está disponível para o tipo `base64Binary`.
- O MTOM pode suportar elementos de anexo dentro de estruturas aninhadas, o que significa que o **bodyPath** para anexos do MTOM são o local **xpath** para o elemento no qual o anexo MTOM é retido. A lógica de computação para **bodyPath** é seguir estritamente o esquema para gerar o local **xpath** conforme mostrado nos exemplos abaixo:
 - Para um tipo de não matriz (**maxOccurs** é 1): `/sendImage/input/imageData`
 - Para um tipo de matriz (**maxOccurs** > 1): `/sendImage/input/imageData[1]`
- Tipos de anexo combinados não são suportados, o que significa que se o MTOM está ativado na ligação de importação, o anexo MTOM será gerado. Se o MTOM está desativado ou se o valor de configuração do MTOM é deixado como o valor padrão na ligação de exportação, a mensagem padrão do MTOM não é suportada.

Anexos referenciados: elementos *swaRef*-typed:

Você pode enviar e receber mensagens SOAP que incluem anexos representados na interface de serviço como elementos com tipo `swaRef`.

Um elemento com tipo `swaRef` é definido no Web Services Interoperability Organization (WS-I) *Attachments Profile* Versão 1.0 (<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), o qual define como elementos de mensagem estão relacionados a partes MIME.

Na mensagem SOAP, o corpo SOAP contém um elemento com tipo `swaRef` que identifica o ID de conteúdo do anexo.

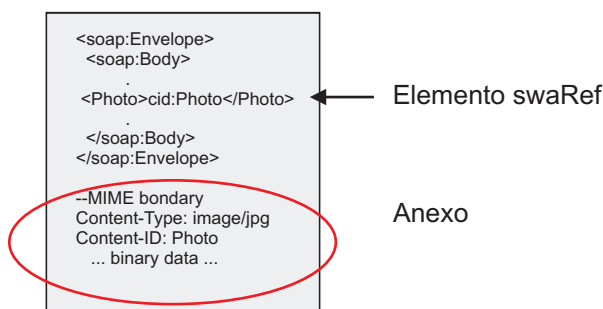


Figura 14. Uma Mensagem SOAP com um Elemento `swaRef`

O WSDL para essa mensagem SOAP contém um elemento com tipo swaRef dentro de uma parte da mensagem que identifica o anexo.

```
<element name="sendPhoto">
  <complexType>
    <sequence>
      <element name="Photo" type="wsi:swaRef"/>
    </sequence>
  </complexType>
</element>
```

O WSDL também deve conter uma ligação MIME que indica que mensagens MIME multipartes devem ser usadas.

Nota: O WSDL *não* inclui uma ligação MIME para o elemento de mensagem com tipo swaRef específico, porque as ligações MIME se aplicam apenas a partes da mensagem de nível superior.

Os anexos representados como elementos com tipo swaRef podem ser propagados apenas entre os componentes do fluxo de mediação. Se um anexo precisar ser acessado ou propagado para outro tipo de componente, use um componente do fluxo de mediação para mover o anexo para um local que seja acessível para esse componente.

Processamento de Entrada de Anexos

Você pode usar o Integration Designer para configurar uma ligação de exportação para receber o anexo. Você cria um módulo e sua interface e operações associadas, incluindo um elemento de tipo swaRef. Você cria uma ligação de serviço da web (JAX-WS).

Nota: Consulte o tópico “Trabalhando com Anexos” no centro de informações do Integration Designer para obter informações mais detalhadas.

Quando um cliente passa uma mensagem SOAP com um anexo swaRef a um componente Service Component Architecture (SCA), a ligação de exportação (JAX-WS) de serviço da web primeiro remove o anexo. Ela, então, analisa a parte SOAP da mensagem e cria um objeto de negócios. Finalmente, a ligação configura o ID de conteúdo do anexo no objeto de negócios.

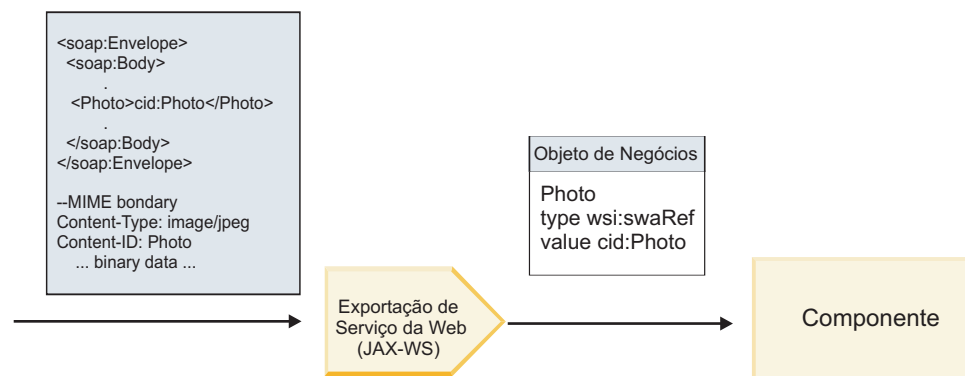


Figura 15. Como a Ligação de Exportação (JAX-WS) de Serviço da Web Processa uma Mensagem SOAP com um Anexo swaRef

Acessando Metadados de Anexo em um Componente do Fluxo de Mediação

Conforme mostrado em Figura 16 na página 88, quando anexos swaRef são acessados por componentes, o identificador do conteúdo do anexo aparece como um elemento de tipo swaRef.

Cada anexo de uma mensagem SOAP também possui um elemento **attachments** correspondente no SMO. Ao usar o tipo swaRef WS-I, o elemento **attachments** inclui o tipo de conteúdo do anexo e o ID do conteúdo bem como os dados binários reais do anexo.

Para obter o valor de um anexo swaRef, é necessário, portanto, obter o valor do elemento com tipo swaRef e, em seguida, localizar o elemento **attachments** com o valor **contentID** correspondente. Observe que o valor **contentID** geralmente tem o prefixo **cid:** removido do valor de swaRef.

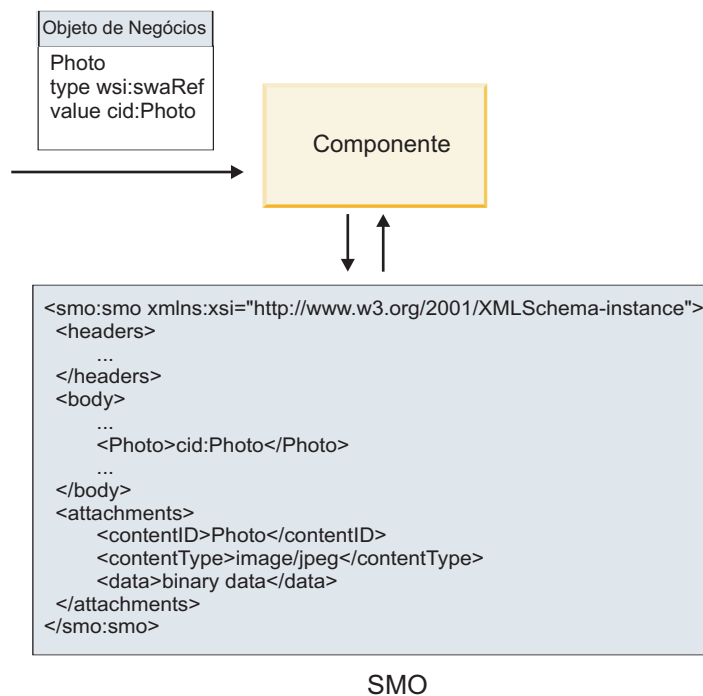


Figura 16. Como Anexos swaRef Aparecem no SMO

Processamento de Saída

Você usa o Integration Designer para configurar uma ligação de importação (JAX-WS) de serviço da web para chamar um serviço da web externo. A ligação de importação é configurada com um documento WSDL que descreve o serviço da web a ser chamado e define o anexo que será passado para o serviço da web.

Quando uma mensagem SCA é recebida por uma ligação de importação (JAX-WS) de serviço da web, os elementos de tipo swaRef são enviados como anexos se a importação estiver ligada a um componente do fluxo de mediação e o elemento de tipo swaRef possuir um elemento correspondente **attachments**.

Para processamento de saída, elementos com tipo swaRef são sempre enviados com seus valores de ID de conteúdo; no entanto, o módulo de mediação deve assegurar que haja um elemento **attachments** correspondente com um valor **contentID** correspondente.

Nota: Para conformidade com o WS-I Attachments Profile, o valor **content ID** deve seguir a "codificação da parte do ID do conteúdo", conforme descrito na seção 3.8 do *WS-I Attachments Profile 1.0*.

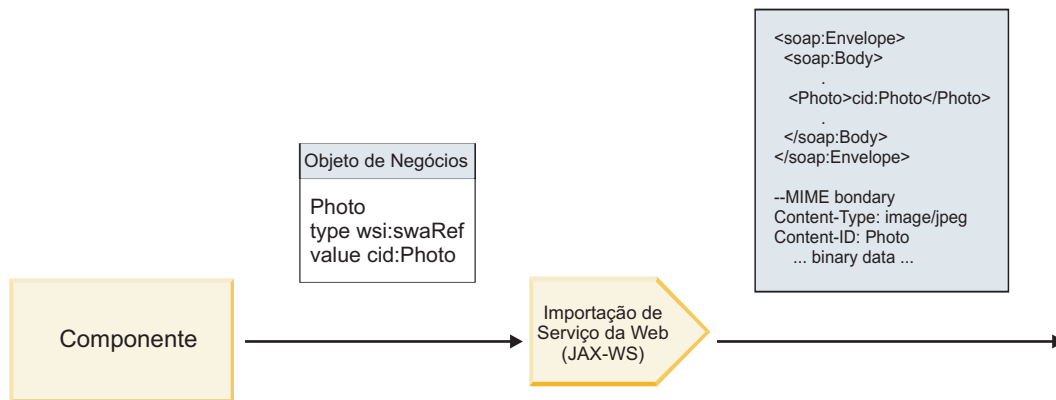
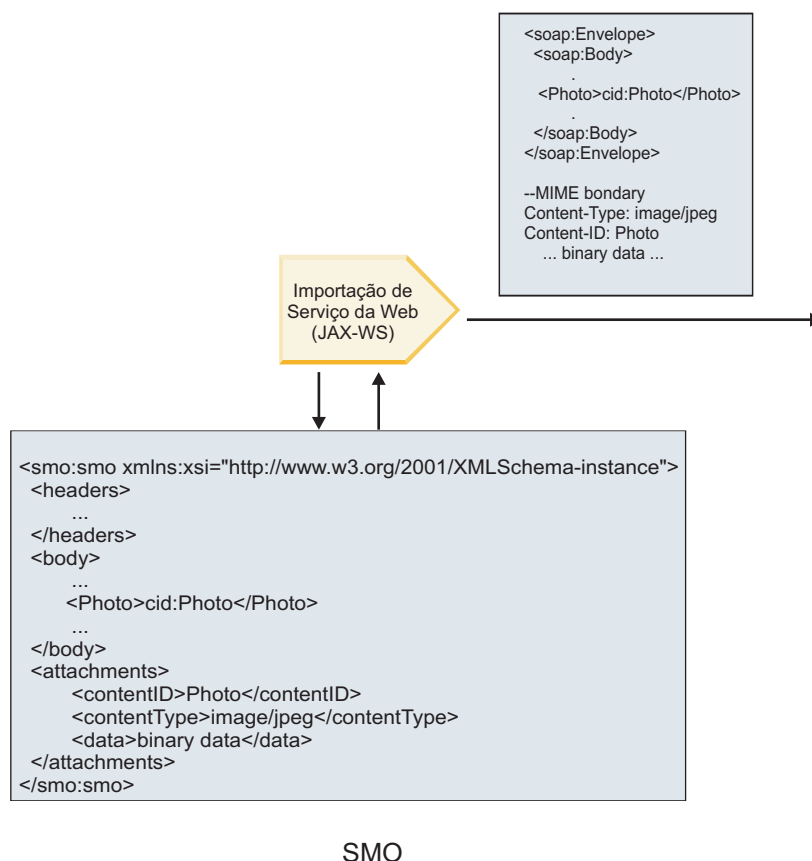


Figura 17. Como a Ligação de Importação (JAX-WS) de Serviço da Web Gera uma Mensagem SOAP com um Anexo swaRef

Configurando Metadados de Anexo em um Componente do Fluxo de Mediação

Se, no SMO, existir um valor de elemento com tipo swaRef e um elemento **attachments**, a ligação preparará a mensagem SOAP (com o anexo) e a enviará para um destinatário.



SMO

Figura 18. Como um Anexo swaRef no SMO é Acessado para Criar a Mensagem SOAP

O elemento **attachments** está presente no SMO apenas se um componente do fluxo de mediação é conectado diretamente à importação ou exportação; ele não é transmitido entre outros tipos de componentes. Se os valores forem necessários em um módulo que contém outros tipos de componentes, um componente do fluxo de mediação deve ser usado para copiar os valores em um local onde eles

podem ser acessados no módulo e outro componente do fluxo de mediação usado para configurar os valores corretos antes de uma chamada de saída por meio de uma importação de serviço da web.

Importante: Como descrito em “Representação XML do SMO,”, a primitiva de mediação Mapeamento transforma mensagens usando uma transformação XSLT 1.0. A transformação opera em uma serialização de XML do SMO. A primitiva de mediação Mapeamento permite que a raiz da serialização seja especificada e o elemento-raiz do documento XML reflita esta raiz.

Quando estiver enviando mensagens SOAP com anexos, o elemento-raiz que você escolheu determinará como os anexos serão propagados.

- Se você usar “/body” como a raiz do mapa XML, todos os anexos serão propagados no mapa por padrão.
- Se você usar “/” como a raiz do mapa, poderá controlar a propagação de anexos.

Anexos referenciados: partes da mensagem em nível superior:

É possível enviar e receber mensagens SOAP que incluem anexos binários que são declarados como partes em sua interface de serviço.

Em uma mensagem SOAP MIME com múltiplas partes, o corpo SOAP é a primeira parte da mensagem e o anexo ou os anexos estão em partes subsequentes.

Qual é a vantagem de enviar ou receber um anexo referenciado em uma mensagem SOAP? Os dados binários que compõem o anexo (que é geralmente muito grande) são mantidos separadamente do corpo da mensagem SOAP para que não precisem ser analisados como XML. Isso resulta em processamento mais eficiente do que se os dados binários fossem mantidos dentro de um elemento XML.

Tipos de Mensagens SOAP com Anexos Referenciados

Começando com a Versão 7.0.0.3 do IBM Business Process Manager, você tem a opção sobre como a mensagem SOAP é gerada:

- **Mensagens Compatíveis com WS-I**

O tempo de execução pode gerar mensagens SOAP que estão em conformidade com os WS-I *Attachments Profile Versão 1.0* e o *WS-I Basic Profile Versão 1.1*. Em uma mensagem SOAP que é compatível com esses perfis, apenas uma parte da mensagem é ligada ao corpo SOAP; para aquelas que são ligadas como anexos, a codificação da parte content-id (conforme descrito no *WS-I Attachments Profile Versão 1.0*) é usada para relacionar o anexo à parte da mensagem.

- **Mensagens Não Compatíveis com WS-I**

O tempo de execução pode gerar mensagens SOAP que não estão em conformidade com os perfis WS-I, mas que são compatíveis com as mensagens geradas na Versão 7.0 ou 7.0.0.2 do IBM Business Process Manager. As mensagens SOAP usam elementos de nível superior denominados após a parte da mensagem com um atributo **href** que contém o anexo **content-id**, mas a codificação da parte content-id (conforme descrito no *WS-I Attachments Profile Versão 1.0*) não é usada.

Selecionando a conformidade de WS-I para exportações de serviço da web

Use o Integration Designer para configurar uma ligação de exportação. Você cria um módulo e sua interface e operações associadas. Você cria uma ligação de serviço da web (JAX-WS). A página Anexos Referenciados exibe todas as partes binárias da operação criada e você pode selecionar quais partes serão anexos. Em seguida, especifique, na página Especifique a Conformidade WS-I AP 1.0 do Integration Designer, uma das seguintes opções:

- **Usar mensagem SOAP compatível com WS-I AP 1.0**

Se você selecionar esta opção, também especificará qual parte da mensagem deve ser ligada ao corpo SOAP.

Nota: Esta opção pode ser usada apenas quando o arquivo WSDL correspondente também é compatível com WS-I.

Um arquivo WSDL que é gerado pelo Integration Designer Versão 7.0.0.3 é compatível com WS-I. No entanto, se você importar um arquivo WSDL que não é compatível com WS-I, não poderá selecionar esta opção.

- **Usar mensagem SOAP não compatível com WS-I AP 1.0**

Se você selecionar esta opção, que é o padrão, a primeira parte da mensagem será ligada ao corpo SOAP.

Nota: Apenas as partes da mensagem de nível superior (ou seja, elementos definidos no portType WSDL como partes dentro da mensagem de entrada ou saída) que possuem um tipo binário (base64Binary ou hexBinary) podem ser enviadas ou recebidas como anexos referenciados.

Consulte o tópico “Trabalhando com Anexos” no centro de informações do Integration Designer para obter informações mais detalhadas.

Para mensagens compatíveis com WS-I, o content-ID que é gerado na mensagem SOAP é uma concatenação dos seguintes elementos:

- O valor do atributo **name** do elemento **wsdl:part** referenciado pelo **mime:content**
- O caractere =
- Um valor globalmente exclusivo, como um UUID
- O caractere @
- Um nome de domínio válido

Processamento de Entrada de Anexos Referenciados

Quando um cliente passa uma mensagem SOAP com um anexo a um componente Service Component Architecture (SCA), a ligação de exportação (JAX-WS) de serviço da web primeiro remove o anexo. Ela, então, analisa a parte SOAP da mensagem e cria um objeto de negócios. Finalmente, a ligação configura o binário de anexo no objeto de negócios.

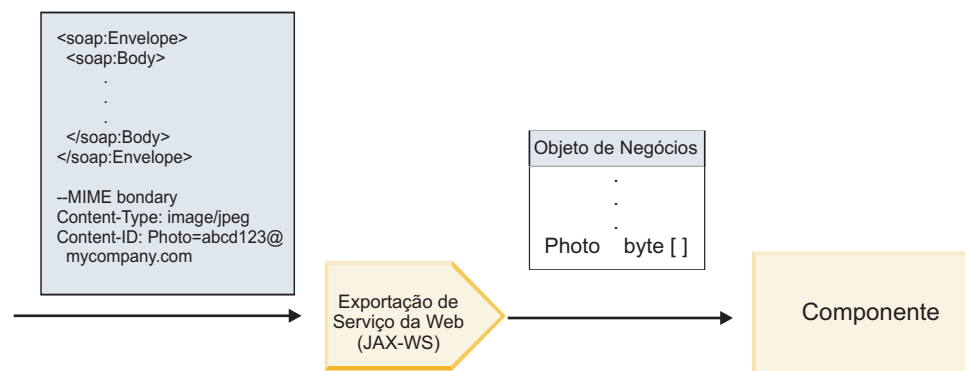


Figura 19. Como a Ligação de Exportação (JAX-WS) de Serviço da Web Processa uma Mensagem SOAP Compatível com WS-I com um Anexo Referenciado

Acessando Metadados de Anexo em um Componente do Fluxo de Mediação

Conforme mostrado em Figura 19, quando anexos referenciados são acessados por componentes, os dados do anexo aparecem como uma matriz de bytes.

Cada anexo referenciado de uma mensagem SOAP também possui um elemento **attachments** correspondente no SMO. O elemento **attachments** inclui o tipo de conteúdo de anexo e o caminho para o

elemento de corpo da mensagem quando o anexo é suspenso.

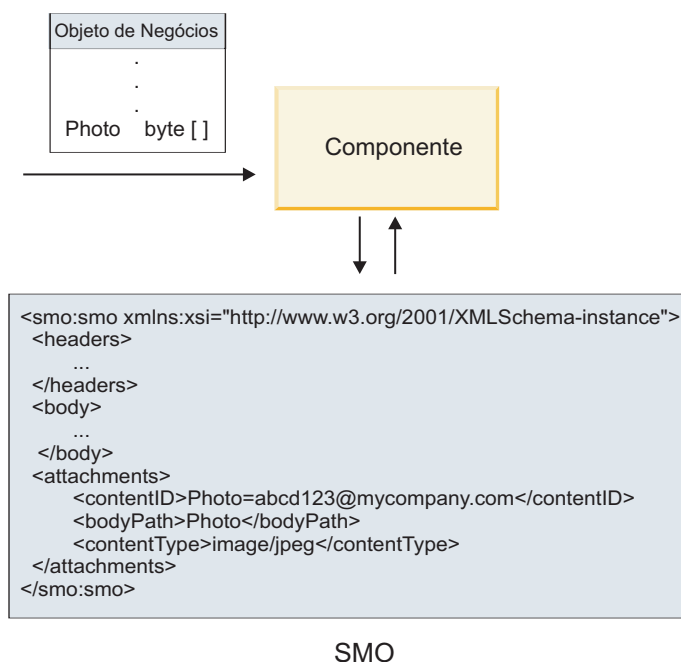


Figura 20. Como Anexos Referenciados Aparecem no SMO

Importante: O caminho para o elemento do corpo da mensagem não é atualizado automaticamente se a mensagem é transformada e o anexo movido. É possível usar um fluxo de mediação para atualizar o elemento **attachments** com o novo caminho (por exemplo, como parte da transformação ou usando um configurador do elemento de mensagem separado).

Como Mensagens SOAP de Saída são Construídas

Você usa o Integration Designer para configurar uma ligação de importação (JAX-WS) de serviço da web para chamar um serviço da web externo. A ligação de importação é configurada com um documento WSDL que descreve o serviço da web a ser chamado e define quais partes da mensagem devem ser passadas como anexos. Também é possível indicar, na página Especifique a Conformidade de WS-I AP 1.0 do Integration Designer, uma das opções a seguir:

- **Usar mensagem SOAP compatível com WS-I AP 1.0**

Se você selecionar esta opção, também especificará qual parte da mensagem deve ser ligada ao corpo SOAP; todas as outras são ligadas a anexos ou cabeçalhos. As mensagens enviadas pela ligação não incluem elementos no corpo SOAP que se referem aos anexos; o relacionamento é expresso por meio do ID de conteúdo do anexo incluindo o nome da parte da mensagem.

- **Usar mensagem SOAP não compatível com WS-I AP 1.0**

Se selecionar esta opção, que é o padrão, a primeira parte da mensagem será ligada ao corpo SOAP; todas as outras serão ligadas aos anexos ou cabeçalhos. As mensagens enviadas pela ligação incluem um ou mais elementos no corpo SOAP que se referem aos anexos por meio de um atributo **href**.

Nota: A parte que representa um anexo, conforme definido no WSDL, deve ser um tipo simples (base64Binary ou hexBinary). Se uma parte for definida por um complexType, essa parte não poderá ser ligada como um anexo.

Processamento de Saída de Anexos Referenciados

A ligação de importação usa informações no SMO para determinar como as partes da mensagem de nível superior binárias são enviadas como anexos.

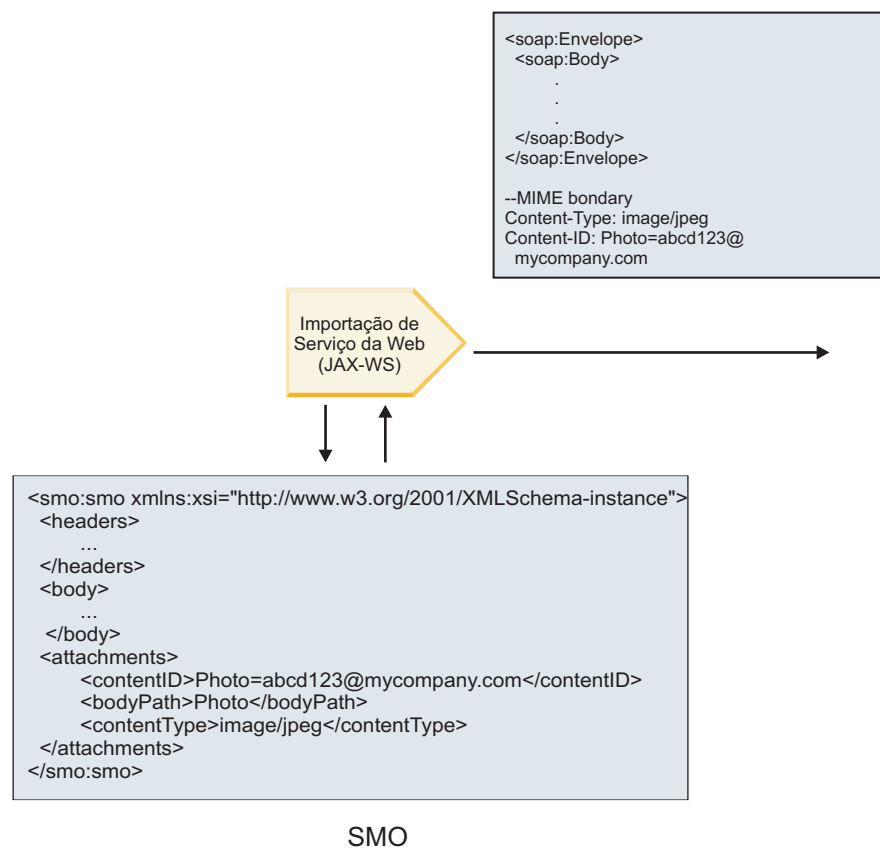


Figura 21. Como o Anexo Referenciado no SMO é Acessado para Criar a Mensagem SOAP

O elemento **attachments** está presente no SMO apenas se um componente do fluxo de mediação é conectado diretamente à importação ou exportação; ele não é transmitido entre outros tipos de componentes. Se os valores forem necessários em um módulo que contém outros tipos de componentes, um componente do fluxo de mediação deve ser usado para copiar os valores em um local onde eles podem ser acessados no módulo e outro componente do fluxo de mediação usado para configurar os valores corretos antes de uma chamada de saída por meio de uma importação de serviço da web.

A ligação usa uma combinação das seguintes condições para determinar como (ou se) a mensagem é enviada:

- Se existe uma ligação MIME WSDL para a parte da mensagem binária de nível superior e, se sim, como o tipo de conteúdo é definido
- Se existe um elemento **attachments** no SMO cujo valor de **bodyPath** referencia uma parte binária de nível superior

Como Anexos são Criados Quando um Elemento attachment Existe no SMO

A tabela a seguir mostra como um anexo é criado e enviado se o SMO contém um elemento **attachment** com um **bodyPath** que corresponde a uma parte do nome da mensagem:

Tabela 32. Como o Anexo é Gerado

Status de ligação MIME WSDL para parte da mensagem binária de nível superior	Como a mensagem é criada e enviada
Presente com um dos seguintes: <ul style="list-style-type: none"> • Nenhum tipo de conteúdo definido para a parte da mensagem • Diversos tipos de conteúdo definidos • Tipo de conteúdo curinga definido 	Parte da mensagem é enviada como um anexo. Content-Id é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, um é gerado. Content-Type é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, é configurado como application/octet-stream .
Presente com conteúdo não curinga único para a parte da mensagem	Parte da mensagem é enviada como um anexo. Content-Id é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, um é gerado. Content-Type é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, é configurado com o tipo definido no elemento de conteúdo MIME WSDL.
Não presente	Parte da mensagem é enviada como um anexo. Content-Id é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, um é gerado. Content-Type é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, é configurado como application/octet-stream . Nota: Enviar partes da mensagem como anexos quando não definido como tal no WSDL pode quebrar a conformidade com o WS-I Attachments Profile 1.0 e, portanto, deve ser evitado, se possível.

Como Anexos são Criados Quando Nenhum Elemento `attachment` Existe no SMO

A tabela a seguir mostra como um anexo é criado e enviado se o SMO não contém um elemento `attachment` com um `bodyPath` que corresponde a uma parte do nome da mensagem:

Tabela 33. Como o Anexo é Gerado

Status de ligação MIME WSDL para parte da mensagem binária de nível superior	Como a mensagem é criada e enviada
Presente com um dos seguintes: <ul style="list-style-type: none"> • Nenhum tipo de conteúdo definido para a parte da mensagem • Diversos tipos de conteúdo definidos • Tipo de conteúdo curinga definido 	Parte da mensagem é enviada como um anexo. Content-Id é gerado. Content-Type é configurado como application/octet-stream .
Presente com conteúdo não curinga único para a parte da mensagem	Parte da mensagem é enviada como um anexo. Content-Id é gerado. Content-Type é configurado com o tipo definido no elemento de conteúdo WSDL MIME.
Não presente	Parte da mensagem não é enviada como um anexo.

Importante: Como descrito em “Representação XML do SMO,”, a primitiva de mediação Mapeamento transforma mensagens usando uma transformação XSLT 1.0. A transformação opera em uma serialização

de XML do SMO. A primitiva de mediação Mapeamento permite que a raiz da serialização seja especificada e o elemento-raiz do documento XML reflita esta raiz.

Quando estiver enviando mensagens SOAP com anexos, o elemento-raiz que você escolheu determinará como os anexos serão propagados.

- Se você usar “/body” como a raiz do mapa XML, todos os anexos serão propagados no mapa por padrão.
- Se você usar “/” como a raiz do mapa, poderá controlar a propagação de anexos.

Anexos não referenciados:

Você pode enviar e receber anexos *não referenciados* que não são declarados na interface de serviço.

Em uma mensagem SOAP MIME com várias partes, o corpo SOAP é a primeira parte da mensagem e os anexos estão em partes subsequentes. Nenhuma referência ao anexo é incluída no corpo SOAP.

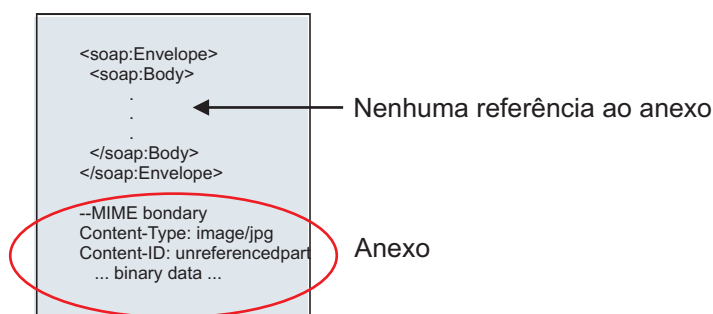


Figura 22. Uma Mensagem SOAP com um Anexo Não Referenciado

É possível enviar uma mensagem SOAP com um anexo não referenciado por meio de uma exportação de serviço da web para uma importação de serviço da web. A mensagem de saída, que é enviada para o serviço da web de destino, contém o anexo.

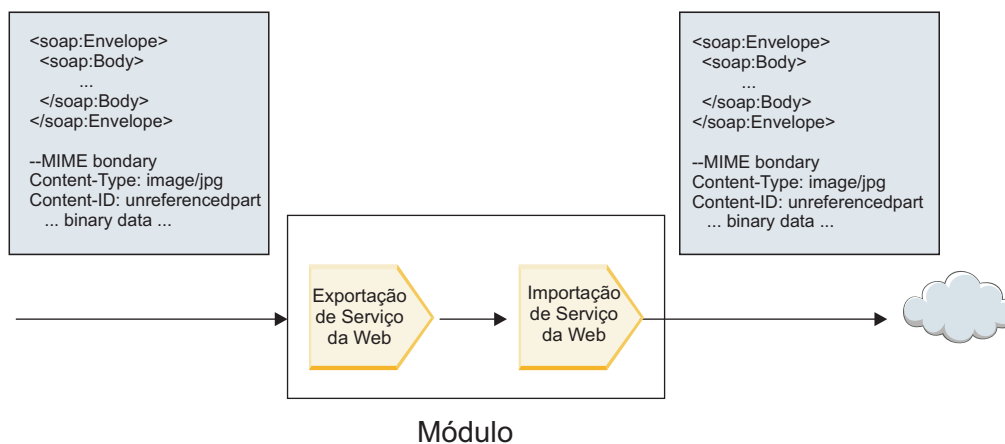


Figura 23. Um Anexo que Passa por meio de um Módulo SCA

No Figura 23, a mensagem SOAP, com o anexo, é transmitida sem modificação.

Também é possível modificar a mensagem SOAP usando um componente de fluxo de mediação. Por exemplo, você pode usar o componente do fluxo de mediação para extrair dados da mensagem SOAP (dados binários no corpo da mensagem, nesse caso) e criar uma mensagem SOAP com anexos. Os dados

são processados como parte do elemento anexos de um objeto de mensagem de serviço (SMO).

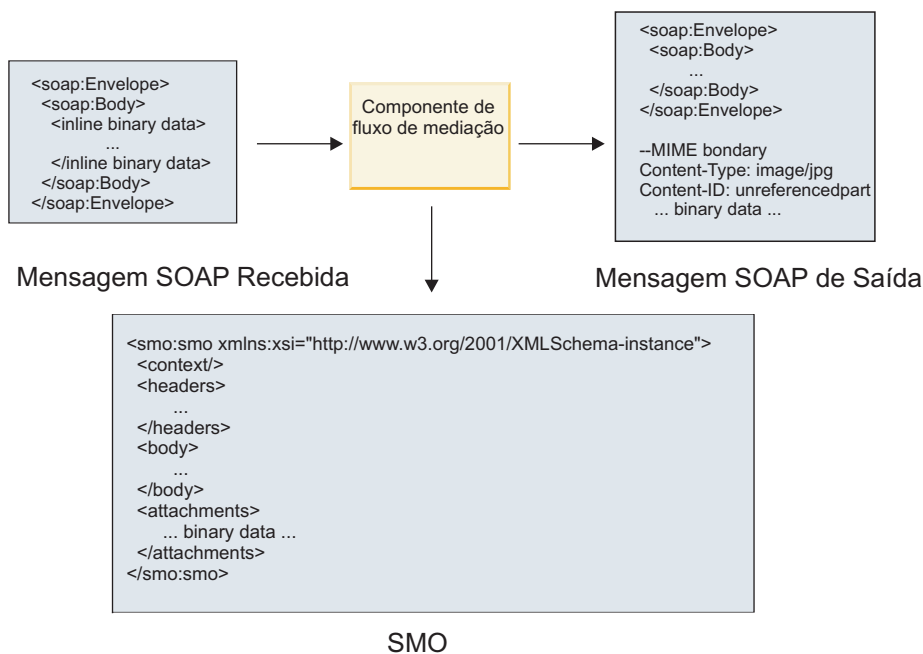


Figura 24. Uma Mensagem Processada por um Componente do Fluxo de Mediação

Em contrapartida, o componente do fluxo de mediação pode transformar a mensagem recebida extraindo e codificando o anexo e, em seguida, transmitindo a mensagem sem anexos.

Em vez de extrair dados de uma mensagem SOAP recebida para formar uma mensagem SOAP com anexos, você pode obter os dados de anexo de uma origem externa, como um banco de dados.

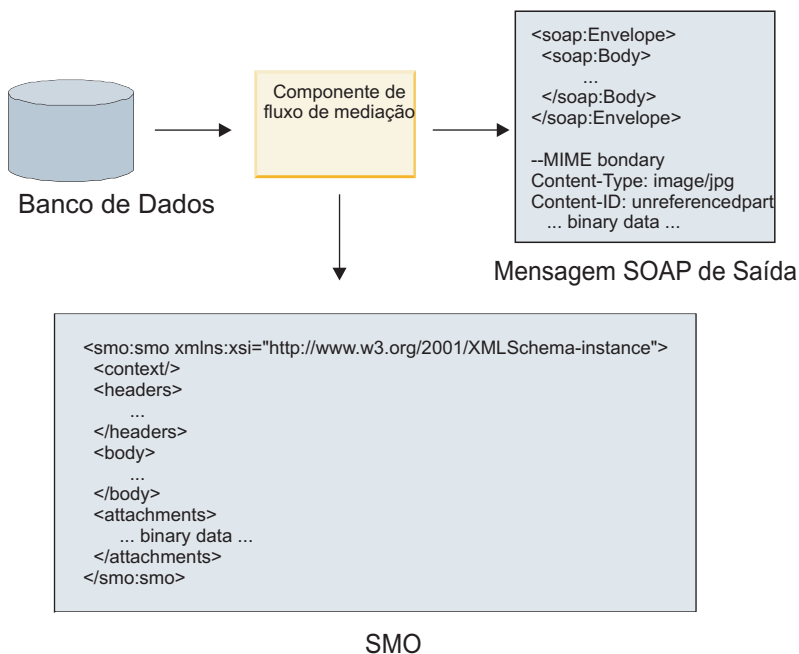


Figura 25. Um Anexo Obtido de um Banco de Dados e Incluído na Mensagem SOAP

De modo inverso, o componente do fluxo de mediação pode extrair o anexo de uma mensagem SOAP recebida e processar a mensagem (por exemplo, armazenar o anexo em um banco de dados).

Anexos não referenciados podem ser propagados apenas entre os componentes do fluxo de mediação. Se um anexo precisar ser acessado ou propagado para outro tipo de componente, use um componente do fluxo de mediação para mover o anexo para um local que seja acessível para esse componente.

Importante: Como descrito em “Representação XML do SMO,”, a primitiva de mediação Mapeamento transforma mensagens usando uma transformação XSLT 1.0. A transformação opera em uma serialização de XML do SMO. A primitiva de mediação Mapeamento permite que a raiz da serialização seja especificada e o elemento-raiz do documento XML reflita esta raiz.

Quando estiver enviando mensagens SOAP com anexos, o elemento-raiz que você escolheu determinará como os anexos serão propagados.

- Se você usar “/body” como a raiz do mapa XML, todos os anexos serão propagados no mapa por padrão.
- Se você usar “/” como a raiz do mapa, poderá controlar a propagação de anexos.

Uso da Ligação de Estilo do Documento WSDL com Mensagens Multipartes:

A organização Web Services Interoperability Organization (WS-I) definiu um conjunto de regras em relação a como os serviços da web devem ser descritos por meio de um WSDL e como as mensagens SOAP correspondentes devem ser formadas a fim de assegurar a interoperabilidade.

Essas regras são especificadas no WS-I *Basic Profile Versão 1.1* (<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>). Em particular, o WS-I Basic Profile 1.1 R2712 afirma: "Uma ligação literal do documento DEVE ser serializada como um ENVELOPE com um soap:Body cujo elemento filho é uma instância da declaração de elemento global referenciada pela parte wsdl:message correspondente".

Isto significa que, ao usar uma ligação SOAP de estilo de documento para uma operação com mensagens (entrada, saída ou falha) que são definidas com diversas partes, apenas uma dessas partes deve ser ligada ao corpo SOAP para ser compatível com o WS-I Basic Profile 1.1.

Além disso, o WS-I Attachments Profile 1.0 R2941 afirma: "Uma wsdl:binding em uma DESCRIÇÃO DEVE ligar cada wsdl:part de uma wsdl:message no wsdl:portType ao qual ele se refere a um soapbind:body, soapbind:header, soapbind:fault, soapbind:headerfault ou mime:content".

Isso significa que, ao usar uma ligação SOAP de estilo de documento para uma operação com mensagens (entrada, saída ou falha) que são definidas com diversas partes, todas as partes diferentes da selecionado a serem ligadas ao corpo SOAP devem ser ligadas como anexos ou cabeçalhos.

A seguinte abordagem é usada quando as descrições WSDL são geradas para exportações com ligações de serviço da web (JAX-WS e JAX-RPC) neste caso:

- É possível escolher qual parte da mensagem é ligada ao corpo SOAP se houver mais de um elemento de tipo não binário. Se houver um único elemento de tipo não binário, esse elemento será automaticamente ligado ao corpo SOAP.
- Para a ligação de JAX-WS, todas as outras partes da mensagem do tipo "hexBinary" ou "Base64Binary" serão ligadas como anexos referenciados. Consulte o “Anexos referenciados: partes da mensagem em nível superior” na página 90.
- Todas as outras partes da mensagem são ligadas como cabeçalhos SOAP.

As ligações de importação JAX-RPC e JAX-WS honram a ligação SOAP em um documento WSDL existente com mensagens de estilo de documento de diversas partes mesmo se não ligar diversas partes ao corpo do SOAP; no entanto, não é possível gerar clientes de serviço da web para documentos WSDL no Rational Application Developer.

Nota: A ligação JAX-RPC não suporta anexos.

O padrão recomendado ao usar mensagens multipartes com uma operação que possui ligação SOAP de estilo de documento é, portanto:

1. Usar o estilo agrupado documento/literal. Nesse caso, as mensagens sempre têm uma única parte; no entanto, os anexos devem ser não referenciados (conforme descrito em “Anexos não referenciados” na página 95) ou com tipo swaRef (conforme descrito em “Anexos referenciados: elementos swaRef-typed” na página 86) neste caso.
2. Usar o estilo RPC/literal. Nesse caso, não há restrições na ligação de WSDL em termos de número de partes ligadas ao corpo SOAP; a mensagem SOAP que resulta sempre possui um único filho que representa a operação que está sendo chamada, com as partes da mensagem sendo filhas desse elemento.
3. Para a ligação de JAX-WS, tenha no máximo uma parte da mensagem que não seja do tipo "hexBinary" ou "Base64Binary", a menos que seja aceitável ligar as outras partes não binárias aos cabeçalhos SOAP.
4. Quaisquer outros casos estão sujeitos ao comportamento descrito.

Nota: Restrições adicionais existem quando você está usando mensagens SOAP que não estão em conformidade com o *WS-I Basic Profile Versão 1.1*.

- A primeira parte da mensagem deve ser não binária.
- Ao receber mensagens SOAP de estilo de documento multipartes com anexos referenciados, a ligação JAX-WS espera que cada anexo referenciado seja representado por um elemento filho do corpo SOAP com um valor de atributo href que identifica o anexo pelo seu ID de conteúdo. A ligação JAX-WS envia anexos referenciados para tais mensagens da mesma maneira. Este comportamento não é compatível com o *WS-I Basic Profile*.

Para assegurar que suas mensagens estejam em conformidade com o *Basic Profile*, siga a abordagem 1 ou 2 na lista anterior ou evite o uso de anexos referenciados para tais mensagens e use anexos não referenciados ou com tipo swaRef em substituição.

Ligações HTTP:

A ligação HTTP é projetada para fornecer conectividade Service Component Architecture (SCA) para HTTP. Consequentemente, aplicativos HTTP existentes ou recém-desenvolvidos podem participar dos ambientes Service Oriented Architecture (SOA).

Hypertext Transfer Protocol (HTTP) é um protocolo amplamente usado para transferir informações na Web. Quando você está trabalhando com um aplicativo externo que usa o protocolo HTTP, uma ligação HTTP é necessária. A ligação HTTP transforma os dados que são passados como uma mensagem no formato nativo para um objeto de negócios em um aplicativo SCA. A ligação HTTP também pode transformar os dados que são atribuídos como um objeto de negócios no formato nativo esperado pelo aplicativo externo.

Nota: Se desejar interagir com clientes e serviços que usam o protocolo SOAP/HTTP de serviços da web, considere usar uma das ligações de serviço da web, que fornecem funcionalidade adicional a respeito da manipulação de qualidades de serviço padrão de serviços da web.

Alguns cenários comuns para usar a ligação HTTP são descritos na lista a seguir:

- Os serviços hospedados pelo SCA podem chamar aplicativos HTTP usando uma importação HTTP.
- Os serviços hospedados pelo SCA podem se expor como aplicativos ativados por HTTP, para que eles possam ser usados pelos clientes HTTP, usando uma exportação HTTP.
- O IBM Business Process Manager e Process Server podem se comunicar entre si por meio de uma infraestrutura de HTTP, consequentemente, os usuários podem gerenciar suas comunicações de acordo com padrões corporativos.

- OIBM Business Process Manager e Process Server podem agir como mediadores de comunicações HTTP, transformação e roteamento de mensagens, o que melhora a integração de aplicativos que usam uma rede HTTP.
- IBM Business Process Manager e Process Server podem ser usados para criar uma ponte entre HTTP e outros protocolos, tais como serviços da Web SOAP/HTTP, adaptadores de recursos baseados em Java Connector Architecture (JCA), JMS e assim por diante.

Informações detalhadas sobre como criar ligações de importação e exportação HTTP podem ser localizadas no centro de informações do Integration Designer. Consulte os tópicos **Desenvolvendo Aplicativos de Integração > Acessando Serviços Externos com HTTP**.

Visão Geral das Ligações HTTP:

A ligação HTTP fornece conectividade para aplicativos hospedados em HTTP. Ela media a comunicação entre aplicativos HTTP e permite que os aplicativos baseados em HTTP existentes sejam chamados a partir de um módulo.

Ligações de Importação HTTP

A ligação de importação HTTP fornece conectividade de saída a partir dos aplicativos Service Component Architecture (SCA) para um servidor HTTP ou aplicativos.

A importação chama uma URL de terminal HTTP. A URL pode ser especificada de uma de três maneiras:

- A URL pode ser configurada dinamicamente nos cabeçalhos HTTP por meio da URL de substituição dinâmica.
- A URL pode ser configurada dinamicamente no elemento de endereço de destino de SMO.
- A URL pode ser especificada como uma propriedade de configuração na importação.

Essa chamada é sempre síncrona por natureza.

Embora chamadas HTTP sempre sejam de solicitação/resposta, a importação de HTTP suporta operações unidirecionais e bidirecionais e ignora a resposta no caso de uma operação unidirecional.

Ligações de Exportação HTTP

A ligação de exportação HTTP fornece conectividade de entrada a partir de aplicativos HTTP para um aplicativo SCA.

Uma URL é definida na exportação HTTP. Aplicativos HTTP que desejam enviar mensagens de solicitação para a exportação usam esta URL para chamar a exportação.

A exportação de HTTP também suporta pings.

Ligações HTTP no Tempo de Execução

Uma importação com uma ligação HTTP no tempo de execução envia uma solicitação com ou sem dados no corpo da mensagem a partir do aplicativo SCA para o serviço da web externo. A solicitação é feita a partir do aplicativo SCA para o serviço da web externo, conforme mostrado no Figura 26 na página 100.

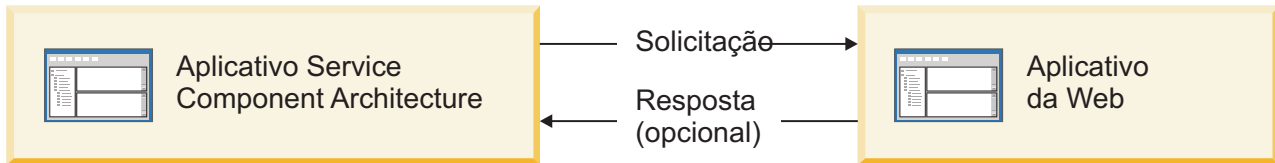


Figura 26. Fluxo de uma Solicitação do Aplicativo SCA para o Aplicativo da Web

Opcionalmente, a importação com a ligação HTTP pode receber dados de volta do aplicativo da web em uma resposta para a solicitação.

Com uma exportação, a solicitação é feita por um aplicativo cliente para um serviço da web, conforme mostrado em Figura 27.

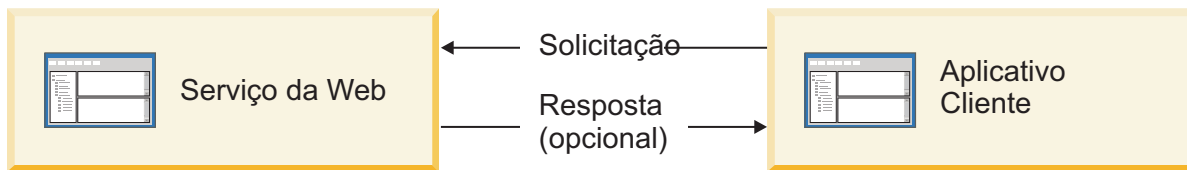


Figura 27. Fluxo de uma solicitação do aplicativo cliente para o serviço da web.

O serviço da Web é um aplicativo da web em execução no servidor. A exportação é implementada nesse aplicativo da Web como um servlet para que o cliente envie sua solicitação para um endereço de URL. O servlet transmite a solicitação para o aplicativo SCA no tempo de execução.

Opcionalmente, a exportação pode enviar dados para o aplicativo cliente em resposta à solicitação.

Cabeçalhos HTTP:

Ligações de importação e exportação HTTP permitem que a configuração de cabeçalhos HTTP e seus valores sejam usados para mensagens de saída. A importação de HTTP usa estes cabeçalhos para solicitações e a exportação de HTTP os usa para respostas.

Os cabeçalhos configurados estaticamente e as informações de controle têm prioridade sobre os valores configurados dinamicamente no tempo de execução. No entanto, os valores de controle da URL, da Versão e do Método de substituição dinâmica substituirão os valores estáticos, que são, de outra forma, considerados os padrões.

A ligação suporta a natureza dinâmica da URL de importação HTTP determinando o valor da URL de destino HTTP, a Versão e o Método no tempo de execução. Estes valores são determinados extraindo o valor de Referência de Terminal, URL de Substituição Dinâmica, Versão e Método.

- Para Referência de Terminal, use as APIs com.ibm.websphere.sca.addressing.EndpointReference ou configure o campo /headers/SMOHeader/Target/address no cabeçalho SMO.
- Para URL de Substituição Dinâmica, Versão e Método, use a seção de parâmetros de controle HTTP da mensagem de Service Component Architecture (SCA). Observe que a URL de Substituição Dinâmica tem precedência sobre a Referência de Terminal de destino; no entanto, a Referência de Terminal se aplica às ligações, por isso, é a abordagem preferencial e deve ser usada quando possível.

As informações de controle e do cabeçalho para mensagens de saída nas ligações de exportação e importação de HTTP são processadas na seguinte ordem:

1. As informações de cabeçalho e controle que excluem a URL de substituição dinâmica HTTP, a Versão e o Método da Mensagem de SCA (prioridade mais baixa)
2. As alterações do console administrativo no nível de exportação/importação

3. As alterações do console administrativo no nível do método da exportação ou importação
4. Endereço de destino especificado pelo caminho da Referência de Terminal ou do cabeçalho SMO
5. URL de Substituição Dinâmica, Versão e Método da mensagem SCA
6. Informações de cabeçalhos e controle do manipulador de dados ou da ligação de dados (prioridade mais alta)

A exportação e importação HTTP irá preencher cabeçalhos de direção de entrada e parâmetros de controle com dados da mensagem de entrada (HTTPExportRequest e HTTPImportResponse) somente se a propagação de cabeçalho de protocolo for configurada como **True**. Inversamente, a exportação e importação HTTP irão ler e processar cabeçalhos de saída e parâmetros de controle (HTTPExportResponse e HTTPImportRequest) somente se a propagação de cabeçalho de protocolo for configurada como **True**.

Nota: As alterações de ligação de dados ou do manipulador de dados para cabeçalhos ou parâmetros de controle na resposta de importação ou solicitação de exportação não alterarão as instruções de processamento da mensagem dentro da ligação de importação ou exportação e devem ser usadas apenas para propagar valores modificados para componentes SCA de recebimento de dados.

O serviço de contexto é responsável por propagar o contexto (incluindo os cabeçalhos de protocolo, tal como o cabeçalho HTTP e o contexto do usuário, tal como o ID da conta) juntamente com um caminho de chamada SCA. Durante o desenvolvimento no IBM Integration Designer, é possível controlar a propagação de contexto por meio de propriedades de importação e exportação. Para obter detalhes adicionais, consulte as informações de ligações de importação e exportação no centro de informações do IBM Integration Designer.

Estruturas de Cabeçalho HTTP e Suporte Fornecidos

O Tabela 34 detalha em itens os parâmetros de solicitação/resposta para solicitações e respostas de Importação HTTP e Exportação HTTP.

Tabela 34. Informações do Cabeçalho HTTP Fornecidas

Nome do Controle	Solicitação de Importação de HTTP	Resposta de Importação de HTTP	Solicitação de Exportação de HTTP	Resposta de Exportação de HTTP
URL	Ignorado	Não Configurado	Ler a partir da mensagem de solicitação. Nota: A sequência de consultas também faz parte do parâmetro de controle da URL.	Ignorado
Versão (valores possíveis: 1.0, 1.1; o padrão é 1.1)	Ignorado	Não Configurado	Ler a partir da mensagem de solicitação	Ignorado
Método	Ignorado	Não Configurado	Ler a partir da mensagem de solicitação	Ignorado

Tabela 34. Informações do Cabeçalho HTTP Fornecidas (continuação)

Nome do Controle	Solicitação de Importação de HTTP	Resposta de Importação de HTTP	Solicitação de Exportação de HTTP	Resposta de Exportação de HTTP
URL de Substituição Dinâmica	Se configurada no manipulador de dados ou na ligação de dados, substitui a URL de Importação HTTP. Gravado na mensagem na linha de solicitação. Nota: A sequência de consultas também faz parte do parâmetro de controle da URL.	Não Configurado	Não Configurado	Ignorado
Versão de Substituição Dinâmica	Se configurada, substitui a Versão de Importação de HTTP. Gravado na mensagem na linha de solicitação.	Não Configurado	Não Configurado	Ignorado
Método de Substituição Dinâmica	Se configurado, substitui o Método de Importação de HTTP. Gravado na mensagem na linha de solicitação.	Não Configurado	Não Configurado	Ignorado
Tipo de Mídia (Este parâmetro de controle transporta parte do valor do cabeçalho HTTP Content-Type.)	Se presente, gravado na mensagem como parte do cabeçalho Content-Type. Nota: Este valor de elemento de controle deve ser fornecido pelo manipulador de dados ou pela ligação de dados.	Ler a partir da mensagem de resposta, cabeçalho Content-Type	Ler a partir da mensagem de solicitação, cabeçalho Content-Type	Se presente, gravado na mensagem como parte do cabeçalho Content-Type. Nota: Este valor de elemento de controle deve ser fornecido pelo manipulador de dados ou pela ligação de dados.
Conjunto de Caracteres (padrão: UTF-8)	Se presente, gravado na mensagem como parte do cabeçalho Content-Type. Nota: Este valor de elemento de controle deve ser fornecido pela ligação de dados.	Ler a partir da mensagem de resposta, cabeçalho Content-Type	Ler a partir da mensagem de solicitação, cabeçalho Content-Type	Suportado; gravado na mensagem como parte do cabeçalho Content-Type. Nota: Este valor de elemento de controle deve ser fornecido pela ligação de dados.
Codificação de Transferência (Valores possíveis : em partes, identidade; o padrão é identidade)	Se presente, gravado na mensagem como um cabeçalho e controla como a transformação da mensagem é codificada.	Leia a partir da mensagem de resposta	Ler a partir da mensagem de solicitação	Se presente, gravado na mensagem como um cabeçalho e controla como a transformação da mensagem é codificada.

Tabela 34. Informações do Cabeçalho HTTP Fornecidas (continuação)

Nome do Controle	Solicitação de Importação de HTTP	Resposta de Importação de HTTP	Solicitação de Exportação de HTTP	Resposta de Exportação de HTTP
Codificação de Conteúdo (Valores possíveis: gzip, x-gzip, deflate, identidade; o padrão é identidade)	Se presente, gravado na mensagem como um cabeçalho e controla como a carga útil é codificada.	Leia a partir da mensagem de resposta	Ler a partir da mensagem de solicitação	Se presente, gravado na mensagem como um cabeçalho e controla como a carga útil é codificada.
Content-Length	Ignorado	Leia a partir da mensagem de resposta	Ler a partir da mensagem de solicitação	Ignorado
StatusCode (padrão: 200)	Não suportado	Leia a partir da mensagem de resposta	Não suportado	Se presente, gravado na mensagem na linha de resposta
ReasonPhrase (padrão: OK)	Não suportado	Leia a partir da mensagem de resposta	Não suportado	Valor de controle ignorado. O valor da linha de resposta da mensagem é gerado a partir do StatusCode.
Autenticação (contém diversas propriedades)	Se presente, usada para construir o cabeçalho de Autenticação Básica. Nota: O valor para este cabeçalho será codificado apenas no protocolo HTTP. No SCA, ele será decodificado e transmitido como texto não criptografado.	Não aplicável	Ler a partir do cabeçalho de Autenticação Básica da mensagem de solicitação. A presença deste cabeçalho não indica que o usuário foi autenticado. A autenticação deve ser controlada na configuração do servlet. Nota: O valor para este cabeçalho será codificado apenas no protocolo HTTP. No SCA, ele será decodificado e transmitido como texto não criptografado.	Não aplicável
Proxy (contém diversas propriedades: Host, Porta, Autenticação)	Se presente, usado para estabelecer a conexão por meio do proxy.	Não aplicável	Não aplicável	Não aplicável
SSL (contém diversas propriedades: Keystore, Senha do Keystore, Trustore, Senha do Trustore, ClientAuth)	Se preenchido e a URL de destino for HTTPS, ele será usado para estabelecer uma conexão por meio de SSL.	Não aplicável	Não aplicável	Não aplicável

Ligações de Dados HTTP:

Para cada mapeamento diferente de dados entre uma mensagem Service Component Architecture (SCA) e uma mensagem de protocolo HTTP, um manipulador de dados ou uma ligação de dados HTTP deve ser configurada. Os manipuladores de dados fornecem uma interface de ligação neutra que permite a reutilização através de ligações de transporte e representam a abordagem recomendada; ligações de dados são específicas para uma ligação de transporte específica. Classes de ligação de dados específicas de HTTP são fornecidas; também é possível gravar manipuladores de dados ou ligações de dados customizadas.

Nota: As três classes de ligação de dados HTTP descritas neste tópico (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML e HTTPServiceGatewayDataBinding) estão descontinuadas a partir do IBM Business Process Manager Versão 7.0. Em vez de usar as ligações de dados descritas neste tópico, considere os manipuladores de dados a seguir:

- Use SOAPDataHandler em vez de HTTPStreamDataBindingSOAP.
- Use UTF8XMLDataHandler em vez de HTTPStreamDataBindingXML
- Use GatewayTextDataHandler em vez de HTTPServiceGatewayDataBinding

As ligações de dados são fornecidas para uso com importações HTTP e exportações HTTP: ligação de dados binários, ligação de dados XML e ligação de dados SOAP. Uma ligação de dados de resposta não é requerida para operações unidirecionais. Uma ligação de dados é representada pelo nome de uma classe Java cujas instâncias podem converter ambos de HTTP para ServiceDataObject e vice-versa. Um seletor de função deve ser usado em uma exportação que, em conjunto com ligações de método, pode determinar qual ligação de método é usada e qual operação é chamada. As ligações de dados fornecidas são:

- Ligações de dados binários, que tratam o corpo como dados binários não estruturados. A implementação do esquema XSD de ligação de dados binários é conforme a seguir:

```
<xsd:schema elementFormDefault="qualified"
  targetNamespace="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:tns="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:complexType name="HTTPBaseBody">
    <xsd:sequence/>
  </xsd:complexType>

  <xsd:complexType name="HTTPBytesBody">
    <xsd:complexContent>
      <xsd:extension base="tns:HTTPBaseBody">
        <xsd:sequence>
          <xsd:element name="value" type="xsd:hexBinary"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

- Ligações de dados XML, que suportam o corpo como dados XML. A implementação da ligação de dados XML é semelhante à ligação de dados XML de JMS e não tem restrições sobre o esquema de interface.
- Ligações de dados SOAP, que suportam o corpo como dados SOAP. A implementação da ligação de dados SOAP não possui restrições sobre o interface de interface.

Implementando Ligações de Dados HTTP Customizadas

Esta seção descreve como implementar uma ligação de dados HTTP customizada.

Nota: A abordagem recomendada é implementar um manipulador de dados customizado porque ele pode ser reutilizado em ligações de transporte.

HTTPStreamDataBinding é a interface principal para manipular mensagens HTTP customizadas. A interface foi projetada para permitir a manipulação de cargas úteis grandes. No entanto, para que essa implementação funcione, essa ligação de dados deve retornar as informações e cabeçalhos de controle antes de gravar a mensagem no fluxo.

Os métodos e suas ordens de execução, listados abaixo, devem ser implementados pela ligação de dados customizada.

Para customizar uma ligação de dados, grave uma classe que implementa HTTPStreamDataBinding. A ligação de dados deve ter quatro propriedades privadas:

- private DataObject pDataObject
- private HTTPControl pCtrl
- private HTTPHeaders pHeaders
- private yourNativeDataType nativeData

A ligação HTTP chamará a ligação de dados customizados na seguinte ordem:

- Processamento de saída (DataObject para o formato Nativo):
 1. setDataObject(...)
 2. setHeaders(...)
 3. setControlParameters(...)
 4. setBusinessException(...)
 5. convertToNativeData()
 6. getControlParameters()
 7. getHeaders()
 8. write(...)
- Processamento de entrada (Formato Nativo para DataObject):
 1. setControlParameters(...)
 2. setHeaders(...)
 3. convertFromNativeData(...)
 4. isBusinessException()
 5. getDataObject()
 6. getControlParameters()
 7. getHeaders()

É necessário chamar setDataObject(...) em convertFromNativeData(...) para configurar o valor de dataObject, o qual é convertido de dado nativo na propriedade privada "pDataObject".

```
public void setDataObject(DataObject dataObject)
    throws DataBindingException {
    pDataObject = dataObject;
}

public void setControlParameters(HTTPControl arg0) {
    this.pCtrl = arg0;
}

public void setHeaders(HTTPHeaders arg0) {
    this.pHeaders = arg0;
}

/*
 * Add http header "IsBusinessException" in pHeaders.
 * Two steps:
 * 1.Remove all the header with name IsBusinessException (case-insensitive) first.
 * This is to make sure only one header is present.
```

```

* 2.Add the new header "IsBusinessException"
*/
public void setBusinessException(boolean isBusinessException) {
    //remove all the header with name IsBusinessException (case-insensitive) first.
    //This is to make sure only one header is present.
    //add the new header "IsBusinessException", code example:
    HTTPHeader header=HeadersFactory.eINSTANCE.createHTTPHeader();
    header.setName("IsBusinessException");
    header.setValue(Boolean.toString(isBusinessException));
    this.pHeaders.getHeader().add(header);
}
public HTTPControl getControlParameters() {
    return pCtrl;
}
public HTTPHeaders getHeaders() {
    return pHeaders;
}
public DataObject getDataObject() throws DataBindingException {
    return pDataObject;
}
/*
* Get header "IsBusinessException" from pHeaders, return its boolean value
*/
public boolean isBusinessException() {
    String headerValue = getHeaderValue(pHeaders,"IsBusinessException");
    boolean result=Boolean.parseBoolean(headerValue);
    return result;
}
public void convertToNativeData() throws DataBindingException {
    DataObject dataObject = getDataObject();
    this.nativeData=realConvertWorkFromSD0ToNativeData(dataObject);
}
public void convertFromNativeData(HTTPInputStream arg0){
    //Customer-developed method to
    //Read data from HTTPInputStream
    //Convert it to DataObject
    DataObject dataobject=realConvertWorkFromNativeDataToSD0(arg0);
    setDataObject(dataobject);
}
public void write(HTTPOutputStream output) throws IOException {
    if (nativeData != null)
        output.write(nativeData);
}

```

Ligações EJB:

Ligações de importação de Enterprise JavaBeans (EJB) permitem que os componentes de Service Component Architecture (SCA) chamem serviços fornecidos pela lógica de negócios Java EE em execução em um servidor Java EE. Ligações de exportação EJB permitem que componentes SCA sejam expostos como Enterprise JavaBeans de forma que a lógica de negócios Java EE possa chamar componentes SCA de outra forma indisponíveis para elas.

Ligações de importação EJB:

Ligações de importação EJB permitem que um módulo SCA chame as implementações EJB especificando a forma com que o módulo de consumo é ligado ao EJB externo. A importação de serviços de uma implementação EJB externa permite que os usuários conectem sua lógica de negócios no ambiente do IBM Business Process Manager e participem de um processo de negócios.

Você pode usar o Integration Designer para criar ligações de importação de EJB. É possível usar um dos seguintes procedimentos para gerar as ligações:

- Criando a importação de EJB usando o assistente de serviço externo

É possível usar o assistente de serviço externo no Integration Designer para construir uma importação de EJB baseada em uma implementação existente. O assistente de serviço externo cria serviços com base nos critérios que você fornece. Ele, então, gera objetos de negócios, interfaces e arquivos de importação com base nos serviços descobertos.

- Criando a importação de EJB usando o editor de montagem

É possível criar uma importação de EJB dentro de um diagrama de conjunto usando o editor de montagem do Integration Designer. Na paleta, é possível usar uma Importação ou usar uma classe Java para criar a ligação EJB.

A importação gerada possui ligações de dados que fazem a conexão Java-WSDL em vez de requerer um componente de ponte Java. É possível ligar diretamente um componente a uma referência de Web Services Description Language (WSDL) na importação de EJB que se comunica com um serviço baseado em EJB usando uma interface Java.

A importação de EJB pode interagir com a lógica de negócios Java EE usando o modelo de programação EJB 2.1 ou o modelo de programação EJB 3.0.

A chamada para a lógica de negócios Java EE pode ser local (apenas para EJB 3.0) ou remota.

- A chamada local é usada quando você deseja chamar a lógica de negócios Java EE que reside no mesmo servidor que a importação.

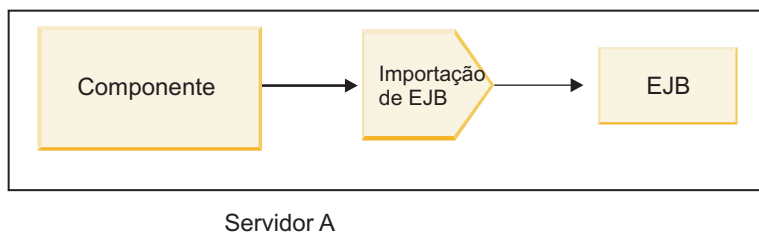


Figura 28. Chamada Local de um EJB (Apenas EJB 3.0)

- A chamada remota é usada quando você deseja chamar a lógica de negócios Java EE que não reside no mesmo servidor que a importação.

Por exemplo, na figura a seguir, uma importação EJB usa o Remote Method Invocation over Internet InterORB Protocol (RMI/IIOP) para chamar um método EJB em outro servidor.

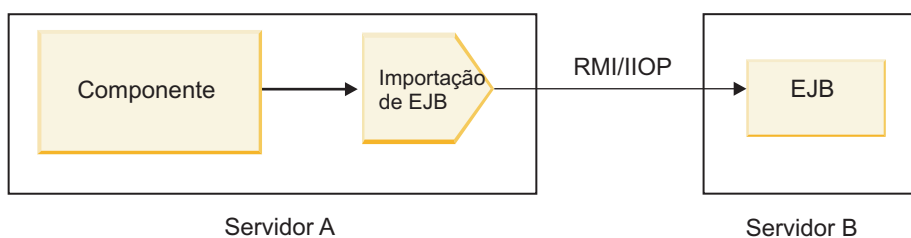


Figura 29. Chamada Remota de um EJB

Quando ele configura a ligação EJB, o Integration Designer usa o nome JNDI para determinar o nível de modelo de programação EJB e o tipo de chamada (local ou remota).

Ligações de importação EJB contêm os seguintes componentes principais:

- Manipulador de dados JAX-WS
- Seletor de falhas EJB
- Seletor de função de importação EJB

Se seu cenário do usuário não for baseado no mapeamento de JAX-WS, você poderá precisar de um manipulador de dados customizado, do seletor de função e do seletor de falha para executar as tarefas de outra forma concluídas pelos componentes que fazem parte das ligações de importação de EJB. Isso inclui o mapeamento normalmente concluído pelo algoritmo de mapeamento customizado.

Ligações de exportação EJB:

Aplicativos Java EE externos podem chamar um componente SCA por meio de uma ligação de exportação EJB. O uso de uma exportação EJB permite que você exponha componentes SCA para que aplicativos Java EE externos possam chamar esses componentes usando o modelo de programação EJB.

Nota: A exportação EJB é um bean stateless.

Use o Integration Designer para criar ligações EJB. É possível usar um dos seguintes procedimentos para gerar as ligações:

- Criando ligações de exportação EJB usando o assistente de serviço externo
É possível usar o assistente de serviço externo no Integration Designer para construir um serviço de exportação de EJB baseado em uma implementação existente. O assistente de serviço externo cria serviços com base nos critérios que você fornece. Em seguida, ele gera objetos de negócios, interfaces e arquivos de exportação com base nos serviços descobertos.
- Criando ligações de exportação EJB usando o editor de montagem
É possível criar uma exportação de EJB usando o editor de montagem do Integration Designer.

Importante: Um cliente Java 2 Platform, Standard Edition (J2SE) não pode chamar o cliente de exportação EJB que é gerado no Integration Designer.

É possível gerar a ligação a partir de um componente SCA existente ou você pode gerar uma exportação com uma ligação EJB para uma interface Java.

- Quando você gera uma exportação para um componente SCA existente que possui uma interface WSDL existente, a exportação tem a uma interface Java designada.
- Quando você gera uma exportação para uma interface Java, é possível selecionar um WSDL ou uma interface Java para a exportação.

Nota: Uma interface Java usada para criar uma exportação de EJB possui as seguintes limitações com relação aos objetos (parâmetros de entrada e saída e exceções) transmitidos como parâmetros em uma chamada remota:

- Eles devem ser do tipo concreto (em vez de uma interface ou tipo abstrato).
- Eles devem estarem conformidade com a especificação do Enterprise JavaBeans. Eles devem ser serializáveis e ter o construtor sem argumento padrão e todas as propriedades devem ser acessíveis por meio dos métodos getter e setter.

Consulte o website da Sun Microsystems, Inc. em <http://java.sun.com> para obter informações sobre a especificação de Enterprise JavaBeans.

Além disso, a exceção deve ser uma exceção verificada, herdada de `java.lang.Exception` e deve ser singular (ou seja, ela não suporta emitir diversos tipos de exceção verificada).

Observe também que a interface de negócios de um Java EnterpriseBean é uma interface Java simples e não deve estender `javax.ejb.EJBObject` ou `javax.ejb.EJBLocalObject`. Os métodos da interface de negócios não deve emitir `java.rmi.remote.exception`.

As ligações de exportação de EJB podem interagir com a lógica de negócios Java EE usando o modelo de programação EJB 2.1 ou o modelo de programação EJB 3.0.

A chamada pode ser local (apenas para EJB 3.0) ou remota.

- A chamada Local é usada quando a lógica de negócios Java EE chama um componente SCA que reside no mesmo servidor que a exportação.
- A chamada remota é usada quando a lógica de negócios Java EE não reside no mesmo servidor que a exportação.

Por exemplo, na figura a seguir, um EJB usa RMI/IIOP para chamar um componente SCA em um servidor diferente.

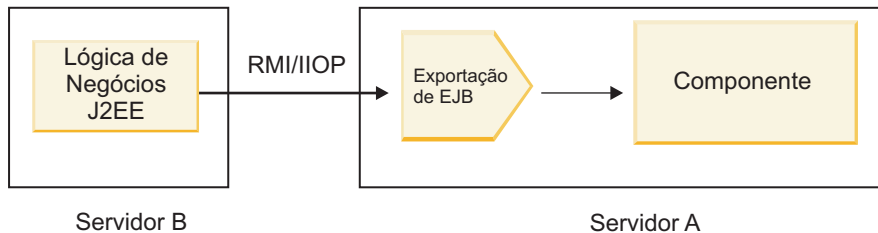


Figura 30. Chamada remota de um cliente para um componente SCA por meio de uma exportação de EJB

Quando ele configura a ligação EJB, o Integration Designer usa o nome JNDI para determinar o nível de modelo de programação EJB e o tipo de chamada (local ou remota).

As ligações de exportação EJB contêm os principais componentes a seguir:

- Manipulador de dados JAX-WS
- Seletor de função de exportação EJB

Se seu cenário do usuário não for baseado no mapeamento de JAX-WS, você poderá precisar de um manipulador de dados customizado e o seletor de função para executar as tarefas de outra forma concluídas pelos componentes que fazem parte das ligações de exportação de EJB. Isso inclui o mapeamento normalmente concluído pelo algoritmo de mapeamento customizado.

Propriedades de ligação EJB:

As ligações de importação EJB usam seus nomes JNDI configurados para determinar o nível de modelo de programação EJB e o tipo de chamada (local ou remoto). As ligações de importação e exportação EJB usam o manipulador de dados JAX-WS para transformação de dados. A ligação de importação EJB usa um seletor de função de importação EJB e um seletor de falhas EJB e a ligação de exportação EJB usa um seletor de função de exportação EJB.

Nomes JNDI e ligações de importação EJB:

Quando ele configura a ligação EJB em uma importação, o Integration Designer usa o nome JNDI para determinar o nível de modelo de programação EJB e o tipo de chamada (local ou remota).

Se nenhum nome JNDI for especificado, a ligação de interface EJB padrão será usada. Os nomes padrão que são criados dependem se você está chamando o EJB 2.1 JavaBeans ou o EJB 3.0 JavaBeans.

Nota: Consulte o tópico "Visão Geral de Ligações do Aplicativo EJB 3.0" no centro de informações do WebSphere Application Server para obter informações detalhadas sobre convenções de nomenclatura.

- JavaBeans EJB 2.1

O nome JNDI padrão pré-selecionado pelo Integration Designer é a ligação do EJB 2.1 padrão, que usa o formato `ejb/` mais a interface inicial, separados por barras.

Por exemplo, para a interface inicial do EJB 2.1 JavaBeans para `com.mycompany.myremotebusinesshome`, a ligação padrão é:

```
ejb/com/mycompany/myremotebusinesshome
```

Para EJB 2.1, apenas a chamada EJB remota é suportada.

- JavaBeans EJB 3.0

O nome JNDI padrão pré-selecionado pelo Integration Designer para o JNDI local é o nome de classe completo da interface local precedido por **ejblocal:**. Por exemplo, para a interface completa da interface `com.mycompany.mylocalbusiness` local, o JNDI EJB 3.0 pré-selecionado é:

```
ejblocal:com.mycompany.mylocalbusiness
```

Para a interface `com.mycompany.myremotebusiness` remota, o JNDI EJB 3.0 pré-selecionado é a interface completa:

```
com.mycompany.myremotebusiness
```

As ligações de aplicativos padrão EJB 3.0 são descritas no local a seguir: visão geral das ligações de aplicativos EJB 3.0. Visão geral de ligações de aplicativos EJB 3.0.

O Integration Designer usará o nome "abreviado" como o local de JNDI padrão para EJBs que usam o modelo de programação da versão 3.0.

Nota: Se a referência JNDI implementada do EJB de destino for diferente do local da ligação JNDI padrão porque um mapeamento customizado foi usado ou configurado, o nome JNDI de destino deverá ser especificado corretamente. É possível especificar o nome no Integration Designer antes da implementação ou, para a ligação de importação, é possível alterar o nome no console administrativo (após a implementação) para corresponder ao nome JNDI do EJB de destino.

Para obter informações adicionais sobre a criação de ligações EJB, consulte a seção dedicada a Trabalhar com Ligações EJB no centro de informações do Integration Designer.

Manipulador de dados JAX-WS:

A ligação de importação do Enterprise JavaBeans (EJB) usa o manipulador de dados do JAX-WS para transformar os objetos de negócios de solicitação em parâmetros de objeto Java e para transformar o valor de retorno do objeto Java no objeto de negócios de resposta. A ligação de exportação do EJB usa o manipulador de dados JAX-WS para transformar EJBs de solicitação nos objetos de negócios de solicitação e para transformar o objeto de negócios de resposta em um valor de retorno.

Esse manipulador de dados mapeia dados da interface WSDL especificada por SCA para a interface Java EJB de destino (e vice-versa) usando a especificação Java API para XML Web Services (JAX-WS) e a especificação Java Architecture para XML Binding (JAXB).

Nota: O suporte atual é restrito às especificações JAX-WS 2.1.1 e JAXB 2.1.3.

O manipulador de dados especificado no nível de ligação EJB é usado para executar o processamento de solicitação, de resposta, de falha e de exceção de tempo de execução.

Nota: Para falhas, um manipulador de dados específico pode ser especificado para cada falha especificando a propriedade de configuração `faultBindingType`. Isso substitui o valor especificado no nível de ligação EJB.

O manipulador de dados JAX-WS é usado por padrão quando a ligação EJB possui uma interface WSDL. Este manipulador de dados não pode ser usado para transformar uma mensagem SOAP que representa uma chamada JAX-WS em um objeto de dados.

A ligação de importação de EJB usa um manipulador de dados para transformar um objeto de dados em uma matriz de Objeto Java (`Object[]`). Durante as comunicações de saída, ocorre o seguinte processamento:

1. A ligação EJB configura o tipo esperado, o elemento esperado e o nome do método de destino no `BindingContext` para corresponder àqueles especificados no WSDL.
2. A ligação EJB chama o método de transformação para o objeto de dados que requer a transformação de dados.

3. O manipulador de dados retorna um `Object[]` que representa os parâmetros do método (na ordem de suas definições no método).
4. A ligação EJB usa o `Object[]` para chamar o método na interface EJB de destino.

A ligação também prepara um `Object[]` para processar a resposta a partir da chamada EJB.

- O primeiro elemento no `Object[]` é o valor de retorno da chamada do método Java.
- Os valores subsequentes representam os parâmetro de entrada para o método.

Isto é requerido para suportar os tipos Entrada/Saída e Saída de parâmetros.

Para parâmetros de tipo Saída, os valores devem ser retornados no objeto de dados de resposta.

O manipulador de dados processa e transforma valores localizados no `Object[]` e, em seguida, retorna uma resposta ao objeto de dados.

O manipulador de dados suporta `xs:AnyType`, `xs:AnySimpleType` e `xs:Any` juntamente com outros tipos de dados XSD. Para ativar o suporte para `xs:Any`, use o **@XmlAnyElement (lax=true)** para a propriedade JavaBeans no código Java, como mostra o seguinte exemplo:

```
public class TestType {
    private Object[] object;

    @XmlAnyElement (lax=true)
    public Object[] getObject() {
        return object;
    }

    public void setObject (Object[] object) {
        this.object=object;
    }
}
```

Isto torna o objeto de propriedade em `TestType` um campo `xs:any`. O valor de classe Java usado no campo `xs:any` deve ter a anotação **@XmlAnyElement**. Por exemplo, se `Address` for a classe Java que está sendo usada para preencher a matriz do objeto, a classe `Address` deverá ter a anotação **@XmlRootElement**.

Nota: Para customizar o mapeamento a partir do tipo XSD para tipos Java definidos pela especificação JAX-WS, altere as anotações JAXB para se ajustarem às suas necessidades comerciais. O manipulador de dados JAX-WS suporta `xs:any`, `xs:anyType` e `xs:anySimpleType`.

As restrições a seguir são aplicáveis para o manipulador de dados JAX-WS:

- O manipulador de dados não inclui suporte para a anotação **@WebParam** do atributo de cabeçalho.
- O namespace para arquivos de esquema de objeto de negócios (arquivos XSD) não inclui mapeamento padrão a partir do nome do pacote Java. A anotação **@XMLSchema** em `package-info.java` também não funciona. A única maneira de criar um XSD com um namespace é usar as anotações **@XmlType** e **@XmlRootElement**. **@XmlRootElement** define o namespace de destino para o elemento global nos tipos do JavaBeans.
- O assistente de importação EJB não cria arquivos XSD para classes não relacionadas. A Versão 2.0 não suporta a anotação **@XmlSeeAlso**, portanto, se a classe-filha não for referida diretamente a partir da classe-pai, um XSD não será criado. A solução para esse problema é executar `SchemaGen` para essas classes-filhas.

`SchemaGen` é um utilitário da linha de comandos (localizado no diretório `WPS_Install_Home/bin`) fornecido para criar os arquivos XSD para um determinado bean. Esses XSDs devem ser copiados manualmente no módulo para que a solução funcione.

Seletor de Falha EJB:

O seletor de falha EJB determina se uma chamada EJB resultou em uma falha, uma exceção de tempo de execução ou uma resposta bem-sucedida.

Se uma falha for detectada, o seletor de falha EJB retornará o nome da falha nativo para o tempo de execução de ligação para que o manipulador de dados JAX-WS possa converter o objeto de exceção em um objeto de negócios de falha.

Em uma resposta bem-sucedida (não falha), a ligação de importação EJB monta uma matriz de objeto Java (Object[]) para retornar os valores.

- O primeiro elemento no Object[] é o valor de retorno da chamada do método Java.
- Os valores subsequentes representam os parâmetro de entrada para o método.

Isto é requerido para suportar os tipos Entrada/Saída e Saída de parâmetros.

Para cenários de exceção, a ligação monta um Object[] e o primeiro elemento representa a exceção emitida pelo método.

O seletor de falha pode retornar qualquer um dos seguintes valores:

Tabela 35. Valores de Retorno

Tipo	Valor de Retorno	Descrição
Falha	ResponseType.FAULT	Retornado quando o Object[] transmitido contém um objeto de exceção.
Exceção de Tempo de Execução	ResponseType.RUNTIME	Retornado se o objeto de exceção não corresponde a nenhum dos tipos de exceção declarada no método.
Resposta Normal	ResponseType.RESPONSE	Retornado em todos os outros casos.

Se o seletor de falha retorna um valor de **ResponseType.FAULT**, o nome de falha nativo é retornado. Este nome de falha nativo é usado pela ligação para determinar o nome da falha WSDL correspondente a partir do modelo e para chamar o manipulador de dados de falha correto.

Seletor de função EJB:

As ligações EJB usam um seletor de função de importação (para processamento de saída) ou um seletor de função de exportação (para processamento de entrada) para determinar o método EJB a chamar.

Seletor de Função de Importação

Para processamento de saída, o seletor de função de importação deriva o tipo de método EJB com base no nome da operação chamada pelo componente SCA que é ligada à importação EJB. O seletor de função procurará a anotação @WebMethod na classe Java anotada do JAX-WS gerado pelo Integration Designer para determinar o nome da operação de destino associada.

- Se a anotação @WebMethod estiver presente, o seletor de função usará a anotação @WebMethod para determinar o mapeamento de método Java correto para o método WSDL.
- Se a anotação @WebMethod estiver ausente, o seletor de função assumirá que o nome do método Java é o mesmo que o nome da operação chamada.

Nota: Este seletor de função é válido apenas para uma interface com tipo WSDL em uma importação EJB, não para uma interface com tipo Java em uma importação EJB.

O seletor de função retorna um objeto java.lang.reflect.Method que representa o método da interface EJB.

O seletor de função usa uma matriz de Objeto Java (Object[]) para conter a resposta do método de destino. O primeiro elemento no Object[] é um método Java com o nome do WSDL e o segundo elemento no Object[] é o objeto de negócios de entrada.

Seletor de Função de Exportação

Para processamento de entrada, o seletor de função de exportação deriva do método de destino a ser chamado a partir do método Java.

O seletor de função de exportação mapeia o nome da operação Java chamado pelo cliente EJB para o nome da operação na interface do componente de destino. O nome do método é retornado como uma Sequência e é resolvido pelo tempo de execução SCA dependendo do tipo de interface do componente de destino.

Ligações EIS:

As ligações de Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Essa comunicação é obtida usando exportações EIS e importações EIS que suportam adaptadores de recursos JCA 1.5 e Websphere Adapters.

Seus componentes SCA podem requerer que dados sejam transferidos para ou a partir de um EIS externo. Ao criar um módulo SCA que requer tal conectividade, inclua (além de seu componente SCA) uma importação ou exportação com uma ligação EIS para comunicação com um EIS externo específico.

Adaptadores de recursos no IBM Integration Designer são usados dentro do contexto de uma importação ou uma exportação. Você desenvolve uma importação ou uma exportação com o assistente de serviço externo e, no seu desenvolvimento, inclui o adaptador de recursos. Uma importação EIS, que permite que seu aplicativo chame um serviço em um sistema EIS, ou uma exportação EIS, que permite que um aplicativo em um sistema EIS chame um serviço desenvolvido no IBM Integration Designer, é criada com um adaptador de recursos. Por exemplo, você poderia criar uma importação com o adaptador JD Edwards para chamar um serviço no sistema JD Edwards.

Quando você usa o assistente de serviço externo, as informações de ligação EIS são criadas para você. Também é possível usar uma outra ferramenta, o editor de montagem, para incluir ou modificar informações sobre ligação. Consulte *Acessando serviços externos com adaptadores* para obter mais informações.

Depois que o módulo SCA que contém a ligação EIS é implementado no servidor, é possível usar o console administrativo para visualizar informações sobre a ligação ou para configurar a ligação.

Visão Geral das Ligações EIS:

A ligação EIS (Enterprise Information System), quando usada com um adaptador de recursos JCA, permite acessar serviços em um Enterprise Information System ou tornar seus serviços disponíveis para o EIS.

O exemplo a seguir mostra como um módulo SCA denominado ContactSyncModule sincroniza informações de contato entre um sistema Siebel e um sistema SAP.

1. O componente SCA denominado ContactSync atende (por meio de uma exportação do aplicativo EIS denominada Contato Siebel) para alterações em contatos Siebel.
2. O componente SCA ContactSync em si faz uso de um aplicativo SAP (por meio de uma importação de aplicativo EIS) para atualizar as informações de contato SAP de acordo.

Como as estruturas de dados usadas para armazenar contatos são diferentes em sistemas Siebel e SAP, o componente SCA ContactSync deve fornecer mapeamento.

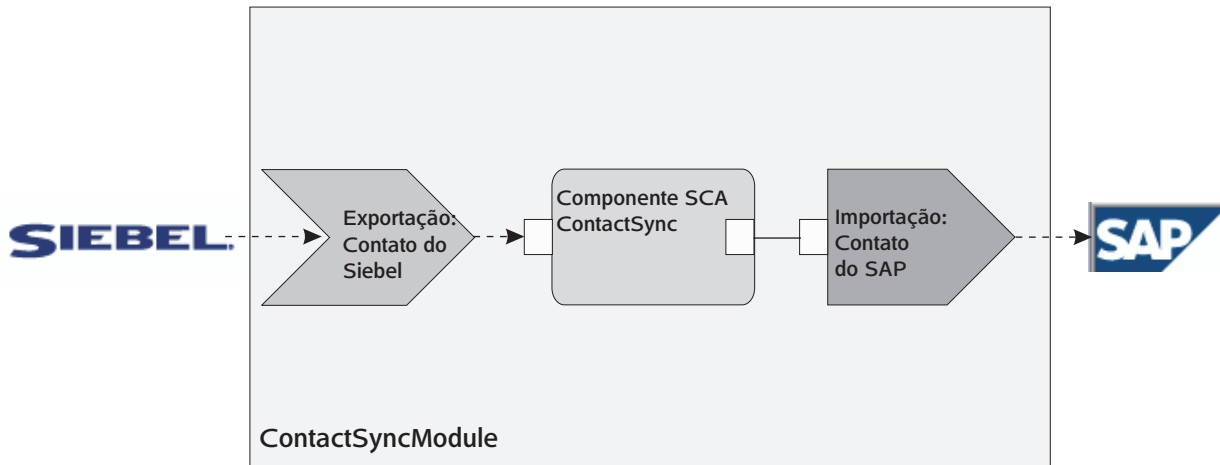


Figura 31. Fluxo a Partir de um Sistema Siebel para um Sistema SAP

A exportação de Contato Siebel e a importação de Contato SAP possuem os adaptadores de recursos apropriados configurados.

Recursos-chave de Ligações EIS:

Uma importação EIS é uma importação de Service Component Architecture (SCA) que permite que os componentes no módulo SCA usem aplicativos EIS definidos fora do módulo SCA. Uma importação EIS é usada para transferir os dados do componente SCA para um EIS externo; uma exportação EIS é usada para transferir dados de um EIS externo para o módulo SCA.

Importações

A função da importação EIS é ligar o intervalo entre os componentes SCA e sistemas EIS externos. Os aplicativos externos podem ser tratados como uma importação EIS. Nesse caso, a importação EIS envia dados para o EIS externo e, opcionalmente, recebe dados em resposta.

A importação EIS fornece componentes SCA com uma visualização uniforme dos aplicativos externos ao módulo. Isso permite que componentes se comuniquem com um EIS externo, tal como SAP, Siebel ou PeopleSoft, usando um modelo SCA consistente.

No lado do cliente da importação, há uma interface, exposta pelo aplicativo de importação EIS, com um ou mais métodos, cada um usando objetos de dados como argumentos e valores de retorno. No lado da implementação, há um Common Client Interface (CCI) implementado por um adaptador de recursos.

A implementação do tempo de execução de uma importação EIS conecta a interface do lado do cliente e o CCI. A importação mapeia a chamada do método na interface para a chamada no CCI.

As ligações são criadas em três níveis: a ligação de interface de ligação, que então usa as ligações de método contidas, que por sua vez usam ligações de dados.

A ligação de interface relaciona a interface da importação à conexão com o sistema EIS que fornece o aplicativo. Isto reflete o fato de que o conjunto de aplicativos, representado pela interface, é fornecido pela instância específica do EIS e a conexão oferece acesso a essa instância. O elemento de ligação contém propriedades com informações suficientes para criar a conexão (estas propriedades são parte da instância `javax.resource.spi.ManagedConnectionFactory`).

A ligação de método associa o método à interação específica com o sistema EIS. Para JCA, a interação é caracterizada pelo conjunto de propriedades da implementação da interface

`javax.resource.cci.InteractionSpec`. O elemento da interação da ligação de método contém estas propriedades, juntamente com o nome da classe, fornecendo, assim, informações suficientes para executar a interação. A ligação de método usa ligações de dados que descrevem o mapeamento do argumento e o resultado do método de interface para representação de EIS.

O cenário de tempo de execução para uma importação EIS é o seguinte:

1. O método na interface de importação é chamado usando o modelo de programação SCA.
2. A solicitação, que atinge a importação EIS, contém o nome do método e seus argumentos.
3. A importação primeiro cria uma implementação de ligação de interface; em seguida, usando dados da ligação de importação, cria uma `ConnectionFactory` e associa as duas. Ou seja, a importação chama `setConnectionFactory` na ligação da interface.
4. A implementação de ligação de método correspondente ao método chamado é criada.
5. A instância `javax.resource.cci.InteractionSpec` é criada e preenchida; em seguida, ligações de dados são usadas para ligar os argumentos do método com um formato entendido pelo adaptador de recursos.
6. A interface CCI é usada para executar a interação.
7. Quando a chamada retorna, a ligação de dados é usada para criar o resultado da chamada e o resultado é retornado para o responsável pela chamada.

Exportações

A função da exportação EIS é ligar o intervalo entre um componente SCA e um EIS externo. Aplicativos externos podem ser tratados como uma exportação EIS. Neste caso, o aplicativo externo envia seus dados na forma de notificações periódicas. Uma exportação EIS pode ser considerada como um aplicativo de assinatura atendendo a uma solicitação externa de um EIS. O componente SCA que usa a exportação EIS a visualiza como um aplicativo local.

A exportação EIS fornece aos componentes SCA uma visualização uniforme dos aplicativos externos ao módulo. Isto permite que componentes se comuniquem com um EIS, tal como SAP, Siebel ou PeopleSoft, usando um modelo SCA consistente.

A exportação apresenta uma implementação do listener recebendo solicitações do EIS. O listener implementa uma interface de listener específica do adaptador de recursos. A exportação também contém uma interface de implementação de componente, exposta ao EIS por meio da exportação.

A implementação do tempo de execução de uma exportação EIS conecta o listener à interface de implementação do componente. A exportação mapeia a solicitação de EIS para a chamada da operação apropriada no componente. As ligações são criadas em três níveis: a ligação de listener, que então usa a ligação de método nativa contida, que por sua vez usa uma ligação de dados.

A ligação do listener relaciona o listener que recebe solicitações com o componente exposto por meio da exportação. A definição de exportação contém o nome do componente; o tempo de execução o localiza e redireciona solicitações para ele.

A ligação de método nativo associa o método nativo ou o tipo de evento recebido pelo listener à operação implementada pelo componente exposto por meio da exportação. Não há relacionamento entre o método chamado no listener e o tipo de evento; todos os eventos chegam por meio de um ou mais métodos do listener. A ligação de método nativo usa o seletor de função definido na exportação para extrair o nome do método nativo a partir dos dados de entrada e ligações de dados para ligar o formato de dados do EIS em um formato entendido pelo componente.

O cenário de tempo de execução para uma exportação EIS é o seguinte:

1. A solicitação de EIS aciona a chamada do método na implementação do listener.
2. O listener localiza e chama a exportação, transmitindo para ela todos os argumentos de chamada.

3. A exportação cria a implementação de ligação do listener.
4. A exportação instancia o seletor de função e o configura na ligação do listener.
5. A exportação inicializa as ligações de método nativo e as inclui na ligação do listener. Para cada ligação de método nativo, as ligações de dados também são inicializadas.
6. A exportação chama a ligação do listener.
7. A ligação do listener localiza componentes exportados e usa o seletor de função para recuperar o nome do método nativo.
8. Este nome é usado para localizar a ligação de método nativo que, então, chama o componente de destino.

O estilo de interação do adaptador permite que a ligação de exportação do EIS chame o componente de destino assincronicamente (o padrão) ou de forma síncrona.

Adaptadores de Recursos

Você desenvolve uma importação ou uma exportação com o assistente de serviço externo e, ao desenvolvê-la, inclui um adaptador de recursos. Os adaptadores que vêm com IBM Integration Designer usados para acessar sistemas Siebel, SAP, PeopleSoft, JD Edwards, IMS e CICS são destinados apenas para propósitos de desenvolvimento e teste. Isso significa que são usados para desenvolver e testar seus aplicativos.

Quando implementar seu aplicativo, adaptadores de tempo de execução licenciados para executar seu aplicativo serão necessários. No entanto, quando constrói seu serviço, é possível incorporar o adaptador com seu serviço. O licenciamento de seu adaptador pode permitir que o adaptador integrado seja usado como o adaptador de tempo de execução licenciado. Estes adaptadores são compatíveis com o Java EE Connector Architecture (JCA 1.5). O JCA, um padrão aberto, é o padrão de Java EE para conectividade do EIS. O JCA fornece uma estrutura gerenciada; isto é, a Qualidade do Serviço (QoS) é fornecida pelo servidor de aplicativos, que oferece gerenciamento de ciclo de vida e segurança a transações. Eles também são compatíveis com a especificação do Enterprise Metadata Discovery, com a exceção do IBM CICS ECI Resource Adapter e do IBM IMS Connector para Java.

Os WebSphere Business Integration Adapters, um conjunto mais antigo de adaptadores, também são suportados pelo assistente.

Recursos Java EE

O módulo EIS, um módulo SCA que segue o padrão de módulo EIS, pode ser implementado na plataforma Java EE.

A implementação do módulo EIS na plataforma Java EE resulta em um aplicativo que está pronto para iniciar, compactado como um arquivo EAR e implementado no servidor. Todos os artefatos Java EE e recursos são criados; o aplicativo está configurado e pronto para ser executado.

Propriedades Dinâmicas de Especificação de Interação JCA e de Especificação de Conexão:

A ligação EIS pode aceitar a entrada para InteractionSpec e ConnectionSpec especificadas usando um objeto de dados filho bem definido que acompanha a carga útil. Isso permite interações de solicitação-resposta dinâmicas com um adaptador de recursos por meio de InteractionSpec e autenticação do componente por meio de ConnectionSpec.

javax.cci.interactionspec transporta informações sobre como a solicitação de interação com o adaptador de recursos deve ser manipulada. Ela também pode transportar informações sobre como a interação foi obtida após a solicitação. Essas comunicações bidirecionais através das interações às vezes são referidas como *conversas*.

A ligação EIS espera que a carga útil que será o argumento para o adaptador de recursos contenha um objeto de dados filho chamado **properties**. Este objeto de dados de propriedade conterá pares nome/valor, com o nome das propriedades de Especificação da Interação em um formato específico. As regras de formatação são:

- Os nomes devem iniciar com o prefixo **IS**, seguido pelo nome da propriedade. Por exemplo, uma especificação de interação com uma propriedade JavaBeans chamada **InteractionId** especificaria o nome da propriedade como **ISInteractionId**.
- O par nome/valor representa o nome e o valor do tipo simples da propriedade de Especificação de Interação.

Neste exemplo, uma interface específica que a entrada de uma operação é um objeto de dados **Account**. Essa interface chama um aplicativo de ligação de importação EIS com a intenção de enviar e receber uma propriedade dinâmica **InteractionSpec** chamada **workingSet** com o valor **xyz**.

O gráfico de negócios ou os objetos de negócios no servidor contêm um objeto de negócios **properties** subjacente que permite o envio de dados específicos de protocolo com a carga útil. Este objeto de negócios **properties** é integrado e não precisa ser especificado no esquema XML ao construir um objeto de negócios. Ele precisa apenas ser criado e usado. Se você tiver seus próprios tipos de dados definidos com base em um esquema XML, será necessário especificar um elemento **properties** que contém seus pares nome/valor esperados.

```
BOFactory dataFactory = (BOFactory) \
serviceManager.locateService("com/ibm/websphere/bo/BOFactory");
//Wrapper for doc-lit wrapped style interfaces,
//skip to payload for non doc-lit
DataObject docLitWrapper = dataFactory.createElement /
("http://mytest/eis/Account", "AccountWrapper");
```

Crie a carga útil.

```
DataObject account = docLitWrapper.createDataObject(0);
DataObject accountInfo = account.createDataObject("AccountInfo");
//Perform your setting up of payload
```

```
//Construct properties data for dynamic interaction
```

```
DataObject properties = account.createDataObject("properties");
```

Para o nome **workingSet**, configure o valor esperado (**xyz**).

```
properties.setString("ISworkingSet", "xyz");
```

```
//Invoke the service with argument
```

```
Service accountImport = (Service) \
serviceManager.locateService("AccountOutbound");
DataObject result = accountImport.invoke("createAccount", docLitWrapper);
```

```
//Get returned property
```

```
DataObject retProperties = result.getDataObject("properties");
```

```
String workingset = retProperties.getString("ISworkingSet");
```

É possível usar propriedades **ConnectionSpec** para autenticação do componente dinâmico. As mesmas regras se aplicam conforme acima, exceto que o prefixo do nome da propriedade precisa ser **CS** (em vez de **IS**). As propriedades **ConnectionSpec** não são bidirecionais. O mesmo objeto de dados **properties** pode conter propriedades **IS** e **CS**.

Para usar propriedades ConnectionSpec, configure o **resAuth** especificado na ligação de importação como **Application**. Além disso, certifique-se de que o adaptador de recursos suporte autorização de componente. Consulte o capítulo 8 da Especificação J2EE Connector Architecture para obter mais detalhes.

Cientes Externos com Ligações EIS:

O servidor pode enviar mensagens para, ou receber mensagens de, clientes externos usando ligações EIS.

Um cliente externo, por exemplo, um portal da web ou um EIS, precisa enviar uma mensagem a um módulo SCA no servidor ou precisa ser chamado por um componente a partir do servidor.

O cliente chama a importação EIS como com qualquer outro aplicativo, usando a Interface de Chamada Dinâmica (DII) ou interface Java.

1. O cliente externo cria uma instância do ServiceManager e consulta a importação EIS usando seu nome de referência. O resultado da consulta é uma implementação da interface de serviço.
2. O cliente cria um argumento de entrada, um objeto de dados genérico, criado dinamicamente usando o esquema de objeto de dados. Esta etapa é feita usando a implementação da interface DataFactory do Objetos de Dados de Serviço.
3. O cliente externo chama o EIS e obtém os resultados requeridos.

Como alternativa, o cliente pode chamar a importação EIS usando a interface Java.

1. O cliente cria uma instância do ServiceManager e consulta a importação de EIS usando seu nome de referência. O resultado da consulta é uma interface Java da importação EIS.
2. O cliente cria um argumento de entrada e um objeto de dados digitado.
3. O cliente chama o EIS e obtém os resultados necessários.

A interface de exportação EIS define a interface do componente SCA exportado que está disponível para os aplicativos EIS externos. Essa interface pode ser considerada como a interface que um aplicativo externo (como SAP ou PeopleSoft) chamará por meio da implementação do tempo de execução do aplicativo de exportação EIS.

A exportação usa EISExportBinding para ligar serviços exportados para o aplicativo EIS externo. Ela permite subscrever um aplicativo contido em seu módulo SCA para atender solicitações de serviço EIS. A ligação de exportação EIS especifica o mapeamento entre a definição de eventos de entrada conforme ele é entendido pelo adaptador de recursos (usando interfaces Java EE Connector Architecture) e a chamada de operações SCA.

EISExportBinding requer que serviços EIS externos sejam baseado em contratos de entrada Java EE Connector Architecture 1.5. EISExportBinding requer que um manipulador de dados ou uma ligação de dados seja especificada no nível de ligação ou no nível de método.

Ligações JMS:

Um provedor de Sistema de Mensagens Java (JMS) ativa o sistema de mensagens com base na API e no modelo de programação do Serviço de Sistema de Mensagens Java. Ele fornece connection factories JMS para criar conexões com destinos JMS e enviar e receber mensagens.

Ligações JMS podem ser usadas ao interagir com a ligação de provedor Barramento de Integração de Serviços (SIB) e são compatíveis com JMS e JCA 1.5.

É possível usar as ligações de exportação e importação JMS um módulo de Service Component Architecture (SCA) para fazer chamadas e receber mensagens de sistemas externos JMS.

As ligações de importação e exportação JMS fornecem integração com aplicativos JMS usando o provedor JMS SIB baseado em JCA 1.5 que faz parte do WebSphere Application Server. Outros adaptadores de recursos JMS baseados em JCA 1.5 não são suportados

Visão Geral de Ligações JMS:

As ligações JMS fornecem conectividade entre o ambiente de Service Component Architecture (SCA) e sistemas JMS.

Ligações JMS

Os principais componentes das ligações de importação JMS e de exportação JMS são:

- adaptador de Recursos: permite a conectividade bidirecional gerenciada entre um módulo SCA e sistemas JMS externos
- Conexões: Encapsular uma conexão virtual entre um cliente e um aplicativo de provedor
- Destinos: usado por um cliente para especificar o destino de mensagens que ele produz ou a origem das mensagens que ele consome
- Dados de Autenticação: usado para proteger o acesso à ligação

Recursos-chave de Ligações JMS

Cabeçalhos Especiais

Propriedades de cabeçalho especiais são usadas em importações e exportações JMS para informar ao destino como manipular a mensagem.

Por exemplo, `TargetFunctionName` mapeia do método nativo para o método de operação.

Recursos Java EE

Um número de recursos Java EE é criado quando importações e exportações do JMS são implementados em um ambiente Java EE.

ConnectionFactory

Usado pelos clientes para criar uma conexão com o provedor JMS.

ActivationSpec

As importações usam isso para receber a resposta para uma solicitação; exportações o usam ao configurar os terminais de mensagem que representam listeners de mensagens em suas interações com o sistema de mensagens.

Destinos

- Destino de envio: em uma importação, este é o local onde a solicitação ou mensagem de saída é enviada; em uma exportação, este é o destino onde a mensagem de resposta será enviada, se não substituído pelo campo de cabeçalho `JMSReplyTo` na mensagem recebida.
- Destino de recebimento: onde a mensagem recebida deve ser colocada; com importações, esta é uma resposta; com exportações, esta é uma solicitação.
- Destino de retorno de chamada: Destino do sistema JMS de SCA usado para armazenar informações de correlação. Não leia ou grave neste destino.

A tarefa de instalação cria `ConnectionFactory` e três destinos. Também cria `ActivationSpec` para ativar o listener de mensagem de tempo de execução para atender respostas no destino de recebimento. As propriedades desses recursos são especificadas no arquivo de importação ou exportação.

Integração de JMS e Adaptadores de Recursos:

O Sistema de Mensagens Java (JMS) fornece integração através de um adaptador de recursos baseado em JMS JCA 1.5 disponível. O suporte completo para integração de JMS é fornecido para o adaptador de recursos do Barramento de Integração de Serviços (SIB) JMS.

Use um provedor JMS para o adaptador de recursos JCA 1.5 quando desejar integrar com um sistema JMS compatível com JCA 1.5 externo. Serviços externos compatíveis com JCA 1.5 podem receber mensagens e enviar mensagens para integrar com seus componentes Service Component Architecture (SCA) usando o adaptador de recursos SIB JMS.

O uso de outros adaptadores de recursos JCA 1.5 específicos do provedor não é suportado.

Ligações de Importação e Exportação JMS:

É possível fazer módulos SCA interagirem com serviços fornecidos por aplicativos JMS externos usando ligações de importação e exportação JMS.

Ligações de Importação JMS

Conexões com o provedor JMS associado de destinos JMS são criadas usando uma connection factory JMS. Use objetos administrativos de connection factory para gerenciar connection factories JMS para o provedor de sistemas de mensagens padrão.

Interação com sistemas JMS externos inclui o uso de destinos para enviar solicitações e receber respostas.

Dois tipos de cenários de uso para a ligação de importação JMS são suportados, dependendo do tipo de operação que está sendo chamado:

- Unidirecional: A importação JMS coloca uma message no destino de envio configurado na ligação de importação. Nada é configurado no campo replyTo do cabeçalho JMS.
- Bidirecional (Solicitação-Resposta): A importação JMS coloca uma mensagem no destino de envio e, em seguida, persiste a resposta que recebe do componente SCA.

A ligação de importação pode ser configurada (usando o campo **Esquema de Correlação de Resposta** no Integration Designer) para esperar que o ID de correlação de mensagem de resposta tenha sido copiado do ID de mensagem de solicitação (o padrão) ou do ID de correlação da mensagem de solicitação. A ligação de importação também pode ser configurada para usar um destino de resposta dinâmica temporária para correlacionar respostas com solicitações. Um destino temporário é criado para cada solicitação e a importação usa este destino para receber a resposta.

O destino de recebimento é configurado na propriedade do cabeçalho replyTo da mensagem de saída. Um listener de mensagem é implementado para atender o destino de recebimento e, quando uma resposta é recebida, o listener de mensagem transmite a resposta de volta para o componente.

Para ambos os cenários de uso, unidirecional e bidirecional, propriedades de cabeçalho dinâmicas e estáticas podem ser especificadas. As propriedades estáticas podem ser configuradas a partir da ligação de método de importação JMS. Algumas destas propriedades tem significados especiais para o tempo de execução JMS SCA.

É importante observar que JMS é uma ligação assíncrona. Se um componente de chamada chamar uma importação JMS sincronicamente (para uma operação bidirecional), o componente de chamada será bloqueado até que a resposta seja retornada pelo serviço JMS.

Figura 32 na página 121 ilustra como a importação está vinculada ao serviço externo.

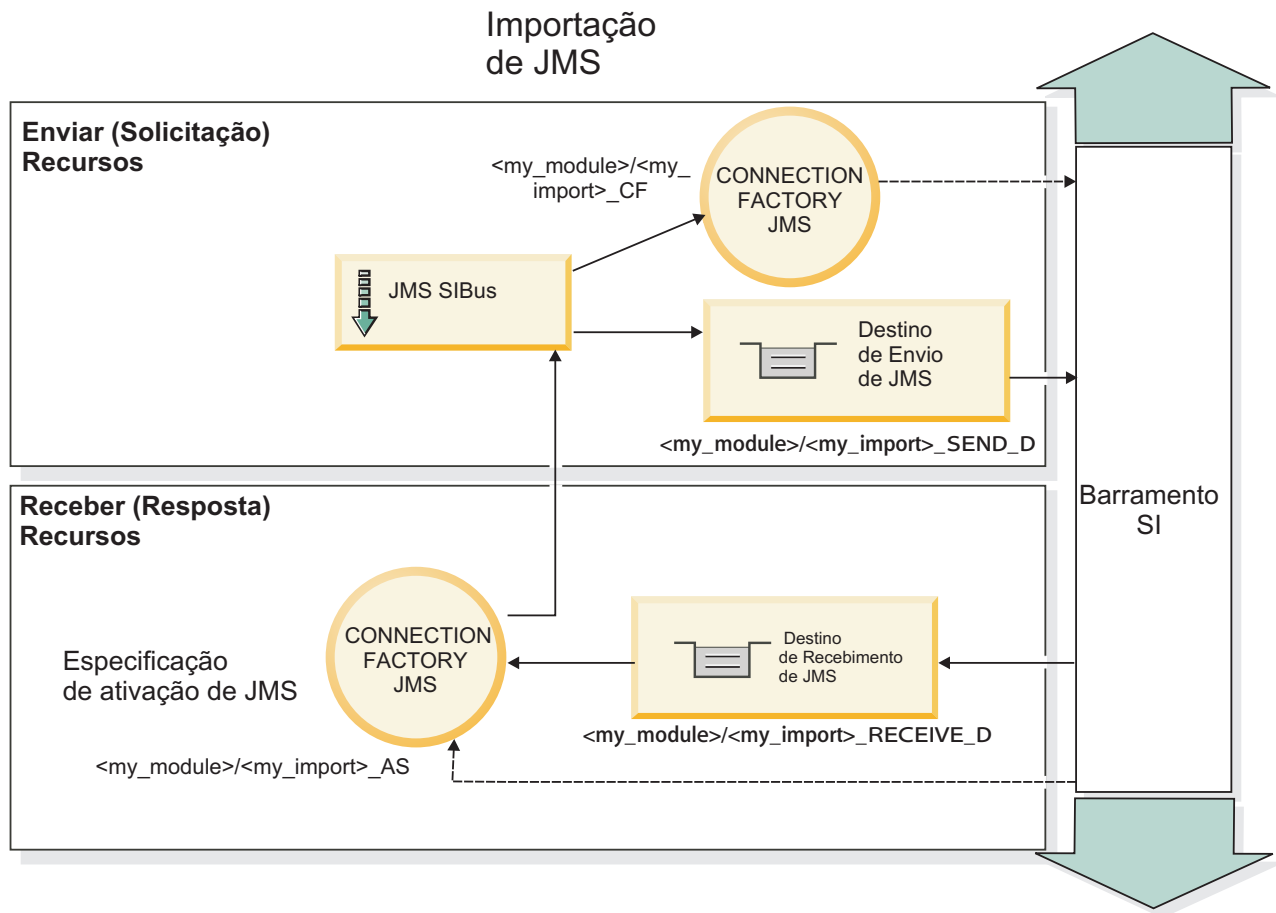


Figura 32. Recursos de Ligação de Importação JMS

Ligações de Exportação JMS

As ligações de exportação JMS fornecem os meios para módulos SCA fornecerem serviços para aplicativos JMS externos.

A conexão que faz parte de uma exportação JMS é uma especificação de ativação configurável.

Uma exportação JMS possui destinos de envio e recebimento.

- O destino de recebimento é onde a mensagem recebida para o componente de destino deve ser colocada.
- O destino de envio é onde a resposta será enviada, a menos que a mensagem recebida tenha substituído isto usando a propriedade de cabeçalho `replyTo`.

Um listener de mensagem é implementado para atender solicitações recebidas no destino de recebimento especificado na ligação de exportação. O destino especificado no campo Enviar é utilizado para enviar a resposta para a solicitação de entrada se o componente chamado fornecer uma resposta. O destino especificado no campo `replyTo` da mensagem recebida substitui o destino especificado no envio.

O Figura 33 na página 122 ilustra como o solicitante externo está vinculado à exportação.

Exportação de JMS

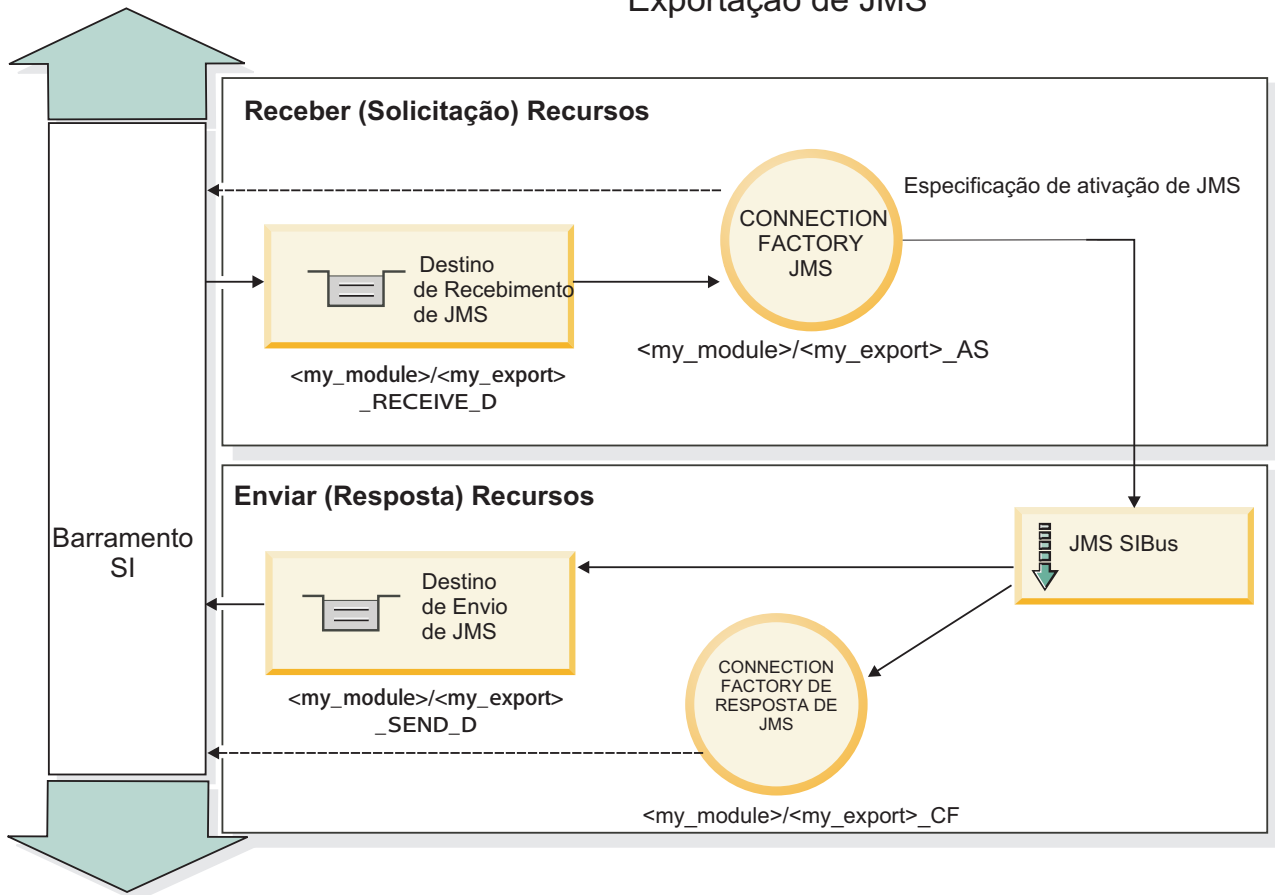


Figura 33. Recursos de Ligação de Exportação JMS

Cabeçalhos JMS:

Uma mensagem JMS contém dois tipos de cabeçalhos - o cabeçalho do sistema JMS e diversas propriedades JMS. Ambos os tipos de cabeçalhos podem ser acessados em um módulo de mediação no Objeto de Mensagem de Serviço (SMO) ou usando a API ContextService.

Cabeçalho do Sistema JMS

O cabeçalho do sistema JMS é representado no SMO pelo elemento JMSHeader, que contém todos os campos geralmente localizados em um cabeçalho JMS. Embora eles possam ser modificados na mediação (ou ContextService), alguns campos de cabeçalho do sistema JMS configurados no SMO não serão propagados na mensagem JMS de saída pois eles são substituídos pelo sistema ou por valores estáticos.

Os campos-chave no cabeçalho do sistema JMS que podem ser atualizados em uma mediação (ou ContextService) são:

- **JMSType** e **JMSCorrelationID** - valores das propriedades de cabeçalho de mensagem predefinidas específicas
- **JMSDeliveryMode** - valores para modo de entrega (persistente ou não persistente; o padrão é persistente)
- **JMSPriority** - valor de prioridade (0 a 9; padrão é JMS_Default_Priority)

Propriedades JMS

As propriedades JMS são representadas no SMO como entradas na lista de Propriedades. As propriedades podem ser incluídas, atualizadas ou excluídas em uma mediação ou usando a API ContextService.

As propriedades também podem ser configuradas estaticamente na ligação JMS. Propriedades que são configuradas estaticamente substituem as configurações (com o mesmo nome) que são configuradas dinamicamente.

As propriedades do usuário propagadas a partir de outras ligações (por exemplo, uma ligação HTTP) serão enviadas na ligação JMS como propriedades JMS.

Configurações de Propagação de Cabeçalho

A propagação do cabeçalho do sistema JMS e das propriedades a partir da mensagem JMS de entrada para componentes de recebimento de dados ou de componentes de envio de dados para a mensagem JMS de saída pode ser controlada pelo sinalizador Propagar Cabeçalho do Protocolo na ligação.

Quando Propagar Cabeçalho de Protocolo é configurado, as informações de cabeçalho têm permitidas para fluírem para a mensagem ou para o componente de destino, conforme descrito na lista a seguir:

- Solicitação de exportação JMS

O cabeçalho JMS recebido na mensagem será propagado para os componentes de destino por meio do serviço de contexto. As propriedades JMS recebidas na mensagem serão propagadas para componentes de destino por meio do serviço de contexto.

- Resposta de exportação JMS

Quaisquer campos de cabeçalho JMS configurados no serviço de contexto serão usados na mensagem de saída, se não substituídos por propriedades estáticas configuradas na ligação de exportação JMS. Quaisquer propriedades configuradas no serviço de contexto serão usadas na mensagem de saída se não substituídas por propriedades estáticas configuradas na ligação de exportação JMS.

- Solicitação de importação JMS

Quaisquer campos de cabeçalho JMS configurados no serviço de contexto serão usados na mensagem de saída, se não substituídos por propriedades estáticas configuradas na ligação de importação JMS. Quaisquer propriedades configuradas no serviço de contexto serão usadas na mensagem de saída se não substituídas por propriedades estáticas configuradas na ligação de importação JMS.

- Resposta de importação JMS

O cabeçalho JMS recebido na mensagem será propagado para os componentes de destino por meio do serviço de contexto. As propriedades JMS recebidas na mensagem serão propagadas para componentes de destino por meio do serviço de contexto.

Esquema de correlação de destino de resposta dinâmica temporária do JMS:

O esquema de correlação de destino de resposta dinâmica temporário faz com que uma fila dinâmica ou tópico exclusivo seja criado para cada solicitação enviada.

O destino da resposta estática especificado na importação é usado para derivar a natureza da fila de destino dinâmica temporária ou tópico. Isto é configurado no campo **ReplyTo** da solicitação e a importação JMS atende respostas nesse destino. Quando a resposta é recebida, ela é reenfileirada para o destino de resposta estática para processamento assíncrono. O campo **CorrelationID** da resposta não é usado e não precisa ser configurado.

Problemas Transacionais

Quando um destino dinâmico temporário está sendo usado, a resposta deve ser consumida no mesmo encadeamento que a resposta enviada. A solicitação deve ser enviada fora da transação global e deve ser confirmada antes de ser recebida pelo serviço de backend e uma resposta retornada.

Persistência

As filas dinâmicas temporárias são entidades de curta duração e não garantem o mesmo nível de persistência associado a uma fila ou tópico estático. Uma fila dinâmica temporária ou tópico não sobreviverá a uma reinicialização de servidor e nem as mensagens. Após a mensagem ter sido reenqueueada no destino de resposta estática, ela retém a persistência definida na mensagem.

Tempo Limite

A importação aguarda para receber a resposta no destino de resposta dinâmica temporário para um período de tempo fixo. Este intervalo de tempo será obtido do qualificador de Prazo de Expiração de Resposta de SCA, se ele estiver configurado, caso contrário, o tempo padrão será 60 segundos. Se o tempo de espera for excedido, a importação lançará uma `IServiceTimeoutRuntimeException`.

Clientes Externos:

O servidor pode enviar mensagens para, ou receber mensagens de, clientes externos usando ligações JMS.

Um cliente externo (como um portal da web ou um Enterprise Information System) pode enviar uma mensagem para um módulo SCA no servidor ou pode ser chamado por um componente a partir do servidor.

Os componentes de exportação JMS implementam listeners de mensagem para atender solicitações recebidas no destino de recebimento especificado na ligação de exportação. O destino especificado no campo de envio é usado para enviar a resposta para a solicitação de entrada se o aplicativo chamado fornece uma resposta. Portanto, um cliente externo é capaz de chamar aplicativos com a ligação de exportação.

As importações JMS interagem com clientes externos enviando mensagens e recebendo mensagens das filas JMS.

Trabalhando com Clientes Externos:

Um cliente externo (ou seja, fora do servidor) pode precisar interagir com um aplicativo instalado no servidor.

Considere um cenário muito simples no qual um cliente externo deseja interagir com um aplicativo no servidor. A figura descreve um cenário simples típico.

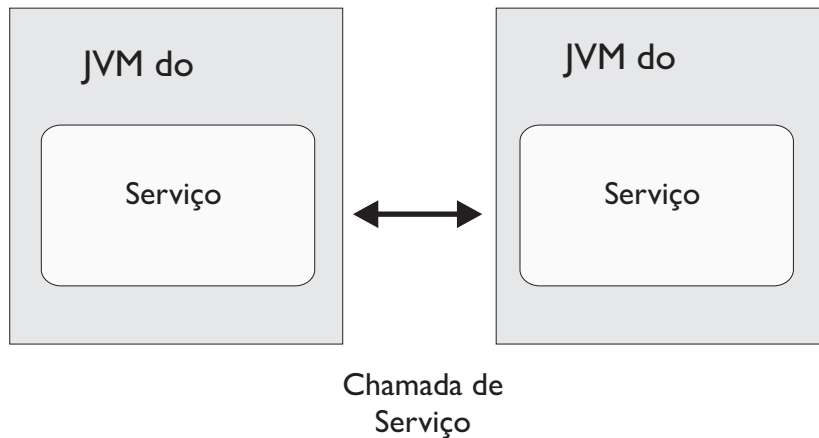


Figura 34. Cenário de Caso de Uso Simples: Cliente Externo Interage com o Aplicativo do Servidor

O aplicativo SCA inclui uma exportação com uma ligação JMS; isso torna o aplicativo disponível para clientes externos.

Quando você tem um cliente externo em uma Java Virtual Machine (JVM) separada de seu servidor, há várias etapas que você deve executar para fazer uma conexão e interagir com uma exportação de JMS. O cliente obtém um InitialContext com os valores corretos e, em seguida, consulta os recursos através de JNDI. O cliente, então, usa o cliente de especificação JMS 1.1 para acessar os destinos e as mensagens de envio e recebimento nos destinos.

Os nomes JNDI padrão dos recursos criados automaticamente pelo tempo de execução são listados no tópico de configuração desta seção. No entanto, se você tiver recursos pré-criados, use esses nomes JNDI.

1. Configure os destinos JMS e a connection factory para enviar a mensagem.
2. Certifique-se de que o contexto JNDI, a porta para o adaptador de recursos do SIB e a porta de autoinicialização do sistema de mensagens estejam corretos.

O servidor usa algumas portas padrão mas, se houver mais servidores instalados nesse sistema, portas alternativas serão criadas no momento da instalação para evitar conflitos com outras instâncias do servidor. É possível usar o console administrativo para determinar quais portas seu servidor está utilizando. Acesse **Servidores > Servidores de Aplicativos > your_server_name > Configuração** e clique em **Portas** em **Comunicação**. É possível, então, editar a porta que está sendo usada.

3. O cliente obtém um contexto inicial com os valores corretos e, em seguida, consulta os recursos através de JNDI.
4. Usando especificações JMS 1.1, o cliente acessa os destinos e as mensagens de envio e recebimento nos destinos.

Resolução de Problemas de Ligações JMS:

É possível diagnosticar e corrigir problemas com ligações JMS.

Exceções de Implementação

Em resposta a várias condições de erro, a implementação de importação e exportação de JMS pode retornar um de dois tipos de exceções:

- Exceção de Negócios de Serviço: esta exceção é retornada se a falha especificada na interface de negócios de serviço (tipo de porta WSDL) ocorreu.
- Exceção de Tempo de Execução de Serviço: levantada em todos os outros casos. Na maioria dos casos, a exceção de causa conterá a exceção original (JMSException).

Por exemplo, uma importação espera apenas uma mensagem de resposta para cada mensagem de solicitação. Se mais de uma resposta chegar, ou se uma resposta atrasada (uma para a qual a expiração de resposta de SCA expirou) chegar, uma Exceção de Tempo de Execução de Serviço será emitida. A transação é retrocedida e a mensagem de resposta é retirada da fila ou manipulada pelo gerenciador de eventos com falha.

Principais Condições de Falha

As principais condições de falha das ligações JMS são determinadas pelas semântica transacionais, pela configuração do provedor JMS ou pela referência ao comportamento existente em outros componentes. As principais condições de falha incluem:

- Falha ao se conectar ao provedor JMS ou destino.

Uma falha de conexão com o provedor JMS para receber mensagens resultará na falha do listener de mensagem ao iniciar. Essa condição será registrada no log do WebSphere Application Server. As mensagens persistentes permanecerão no destino até que elas sejam recuperadas com sucesso (ou expiradas).

Uma falha de conexão com o provedor JMS para enviar mensagens de saída resultará no retrocesso da transação que controla o envio.

- Falha ao analisar uma mensagem de entrada ou ao construir uma mensagem de saída.

Uma falha na ligação de dados ou no manipulador de dados causa o retrocesso da transação que controla o trabalho.

- Falha ao enviar a mensagem de saída.

Uma falha ao enviar uma mensagem causa o retrocesso da transação relevante.

- Mensagens de resposta atrasadas diversas ou inesperadas.

A importação espera apenas uma mensagem de resposta para cada mensagem de solicitação. Além disso, o período de tempo válido no qual uma resposta pode ser recebida é determinado pelo qualificador de Expiração de Resposta SCA na solicitação. Quando uma resposta chega ou o tempo de expiração é excedido, o registro de correlação é excluído. Se mensagens de resposta chegarem inesperadamente ou chegarem atrasadas, uma Exceção de Tempo de Execução de Serviço será emitida.

- Exceção de tempo de execução de tempo limite de serviço causada por resposta atrasada ao usar o esquema de correlação de destino de resposta dinâmica temporário.

A importação de JMS expirará após um período de tempo determinado pelo qualificador de expiração de resposta de SCA ou se, se isto não for configurado, será padronizado como 60 segundos.

Mensagens SCA Baseadas em JMS que Não Aparecem no Gerenciador de Eventos com Falha

Se mensagens SCA originadas por uma interação do JMS falharem, você pode encontrar essas mensagens no gerenciador de eventos com falha. Se tais mensagens não estiverem aparecendo no gerenciador de eventos com falha, assegure-se de que o destino de SIB subjacente do destino JMS tenha um valor máximo de entregas com falha maior que **1**. Configurar esse valor como **2** ou mais permite a interação com o gerenciador de eventos com falha durante chamadas de SCA para as ligações JMS.

Exceções de Manipulação:

A maneira na qual a ligação é configurada determina como as exceções que são levantadas por manipuladores de dados ou ligações de dados são manipuladas. Além disso, a natureza do fluxo de mediação determina o comportamento do sistema quando uma exceção desse tipo é lançada.

Uma variedade de problemas pode ocorrer quando um manipulador de dados ou uma ligação de dados é chamada por sua ligação. Por exemplo, um manipulador de dados pode receber uma mensagem que possui uma carga útil corrompida ou pode tentar ler uma mensagem que possua um formato incorreto.

A maneira que sua ligação trata uma exceção desse tipo é determinada por como você implementa o manipulador de dados ou a ligação de dados. O comportamento recomendado é que você projete sua ligação de dados para lançar uma **DataBindingException**.

Quando qualquer exceção de tempo de execução, incluindo uma **DataBindingException**, é lançada:

- Se o fluxo de mediação for configurado para ser transacional, a Mensagem JMS, por padrão, será armazenada no Gerenciador de Evento com Falha para exclusão ou reprodução manual.

Nota: É possível alterar o modo de recuperação na ligação de modo que a mensagem seja retrocedida ao invés de ser armazenada no Gerenciador de Evento com Falha.

- Se o fluxo de mediação não for transacional, a exceção será registrada e a mensagem será perdida.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, qualquer exceção do manipulador de dados é agrupada em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Ligações JMS Genéricas:

A ligação JMS genérica fornece conectividade com provedores compatíveis com JMS 1.1 de terceiros. A operação das ligações JMS Genéricas é semelhante àquela de ligações JMS.

O serviço fornecido por meio de uma ligação JMS permite que um módulo Service Component Architecture (SCA) faça chamadas ou receba mensagens de sistemas externos. O sistema pode ser um sistema JMS externo.

A Ligação JMS Genérica fornece integração com provedores JMS não compatíveis com JCA 1.5 que suportam JMS 1.1 e implementam o Application Server Facility do JMS opcional. A ligação JMS Genérica suporta esses provedores JMS (incluindo Oracle AQ, TIBCO, SonicMQ, WebMethods e BEA WebLogic) que não suportam JCA 1.5, mas suporta o Application Server Facility da especificação JMS 1.1. O provedor JMS integrado pelo WebSphere (SIBJMS), que é um provedor JMS JCA 1.5, não é suportado por esta ligação; ao usar esse provedor, use o "Ligações JMS" na página 118.

Use esta ligação Genérica durante a integração com um sistema baseado em JMS não compatível com JCA 1.5 em um ambiente SCA. Os aplicativos externos de destino podem, então, receber mensagens e enviar mensagens para integrar com um componente SCA.

Visão Geral de Ligações JMS Genéricas:

Ligações JMS genéricas são ligações JMS não JCA que fornecem conectividade entre o ambiente Service Component Architecture (SCA) e sistemas de JMS que estão em conformidade com o JMS 1.1 e que implementam o Application Server Facility do JMS opcional.

Ligações JMS Genéricas

Os principais aspectos de ligações de importação e exportação do JMS Genérico incluem o seguinte:

- Porta Listener: permite que os provedores JMS não baseados em JCA recebam mensagens e as despachem para um Bean Acionado por Mensagens (MDB)
- Conexões: Encapsular uma conexão virtual entre um cliente e um aplicativo de provedor
- Destinos: usado por um cliente para especificar o destino de mensagens que ele produz ou a origem das mensagens que ele consome
- Dados de Autenticação: usado para proteger o acesso à ligação

Ligações de Importação JMS Genéricas

As ligações de importação do JMS genérico permitem que componentes dentro do módulo SCA se comuniquem com serviços fornecidos pelos provedores externos JMS não compatíveis com JCA 1.5.

A parte de conexão de uma importação de JMS é um connection factory. Uma connection factory, o objeto que um cliente usa para criar uma conexão com um provedor, encapsula um conjunto de parâmetros de configuração de conexão definidos por um administrador. Cada connection factory é uma instância da interface ConnectionFactory, QueueConnectionFactory ou TopicConnectionFactory.

Interação com sistemas JMS externos inclui o uso de destinos para enviar solicitações e receber respostas.

Dois tipos de cenários de uso para a ligação de importação do JMS Genérico são suportados, dependendo do tipo de operação que está sendo chamado:

- Unidirecional: A importação do JMS Genérico coloca uma mensagem no destino de envio configurado na ligação de importação. Nada é enviado para o campo replyTo do cabeçalho JMS.
- Bidirecional (Solicitação-Resposta): A importação do JMS Genérico coloca uma mensagem no destino de envio e, em seguida, persiste a resposta que recebe do componente SCA.

O destino de recebimento é configurado na propriedade do cabeçalho replyTo da mensagem de saída. Um bean acionado por mensagens (MDB) é implementado para atender no destino de recebimento e, quando uma resposta é recebida, o MDB transmite a resposta de volta para o componente.

A ligação de importação pode ser configurada (utilizando o campo **Esquema de Correlação de Resposta** no Integration Designer) para esperar que o ID de correlação da mensagem de resposta tenha sido copiado do ID de mensagem de solicitação (o padrão) ou a partir do ID de correlação da mensagem de solicitação.

Para ambos os cenários de uso, unidirecional e bidirecional, propriedades de cabeçalho dinâmicas e estáticas podem ser especificadas. As propriedades estáticas podem ser configuradas a partir da ligação do método de importação do JMS Genérico. Algumas destas propriedades tem significados especiais para o tempo de execução JMS SCA.

É importante observar que JMS Genérica é uma ligação assíncrona. Se um componente de chamada chamar uma importação JMS Genérica sincronicamente (para uma operação bidirecional), o componente de chamada será bloqueado até a resposta ser retornada pelo serviço JMS.

Figura 35 na página 129 ilustra como a importação está vinculada ao serviço externo.

(Genérico) Importação de JMS

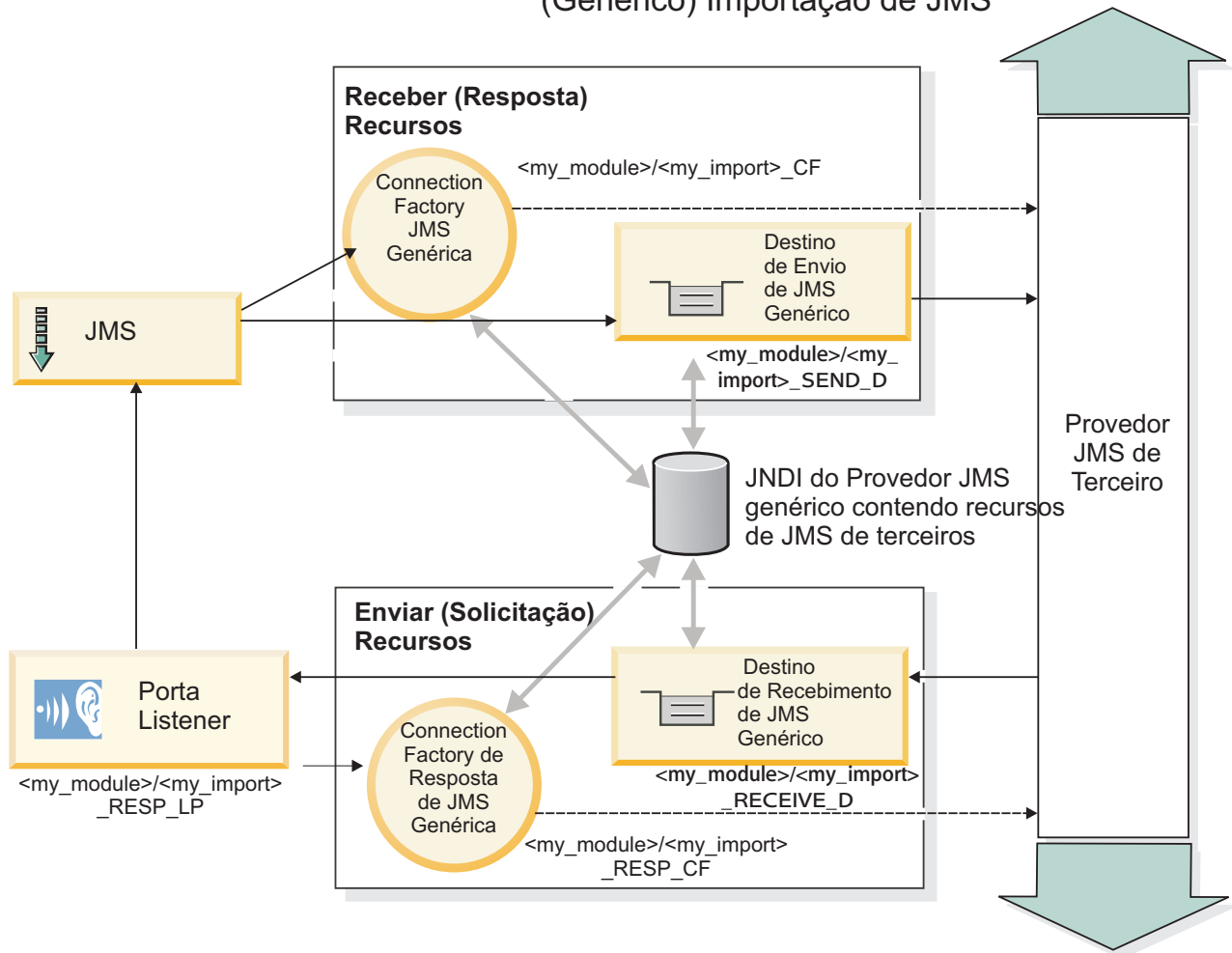


Figura 35. Recursos de Ligação de Importação JMS Genérica

Ligações de Exportação JMS Genéricas

As ligações de exportação JMS genéricas fornecem o meio para módulos SCA fornecerem serviços para aplicativos JMS externos.

A parte da conexão de uma exportação JMS é composta de uma `ConnectionFactory` e uma `ListenerPort`.

Uma exportação do JMS Genérico possui destinos de envio e de recebimento.

- O destino de recebimento é onde a mensagem recebida para o componente de destino deve ser colocada.
- O destino de envio é onde a resposta será enviada, a menos que a mensagem recebida tenha substituído isso usando a propriedade do cabeçalho `replyTo`.

Um MDB é implementado para atender a solicitações recebidas para o destino de recebimento especificado na ligação de exportação.

- O destino especificado no campo de envio será usado para enviar a resposta para a solicitação de entrada, se o componente chamado fornecer uma resposta.
- O destino especificado no campo `replyTo` da mensagem recebida substitui o destino especificado no campo de envio.

- Para cenários de solicitação/resposta, a ligação de importação pode ser configurada (usando o campo **Esquema de Correlação de Resposta** no Integration Designer) para esperar a resposta copiar o ID da mensagem de solicitação no campo ID de Correlação da mensagem de resposta (padrão) ou a resposta pode copiar o ID de correlação no campo ID de Correlação da mensagem de resposta.

O Figura 36 ilustra como o solicitante externo está vinculado à exportação.

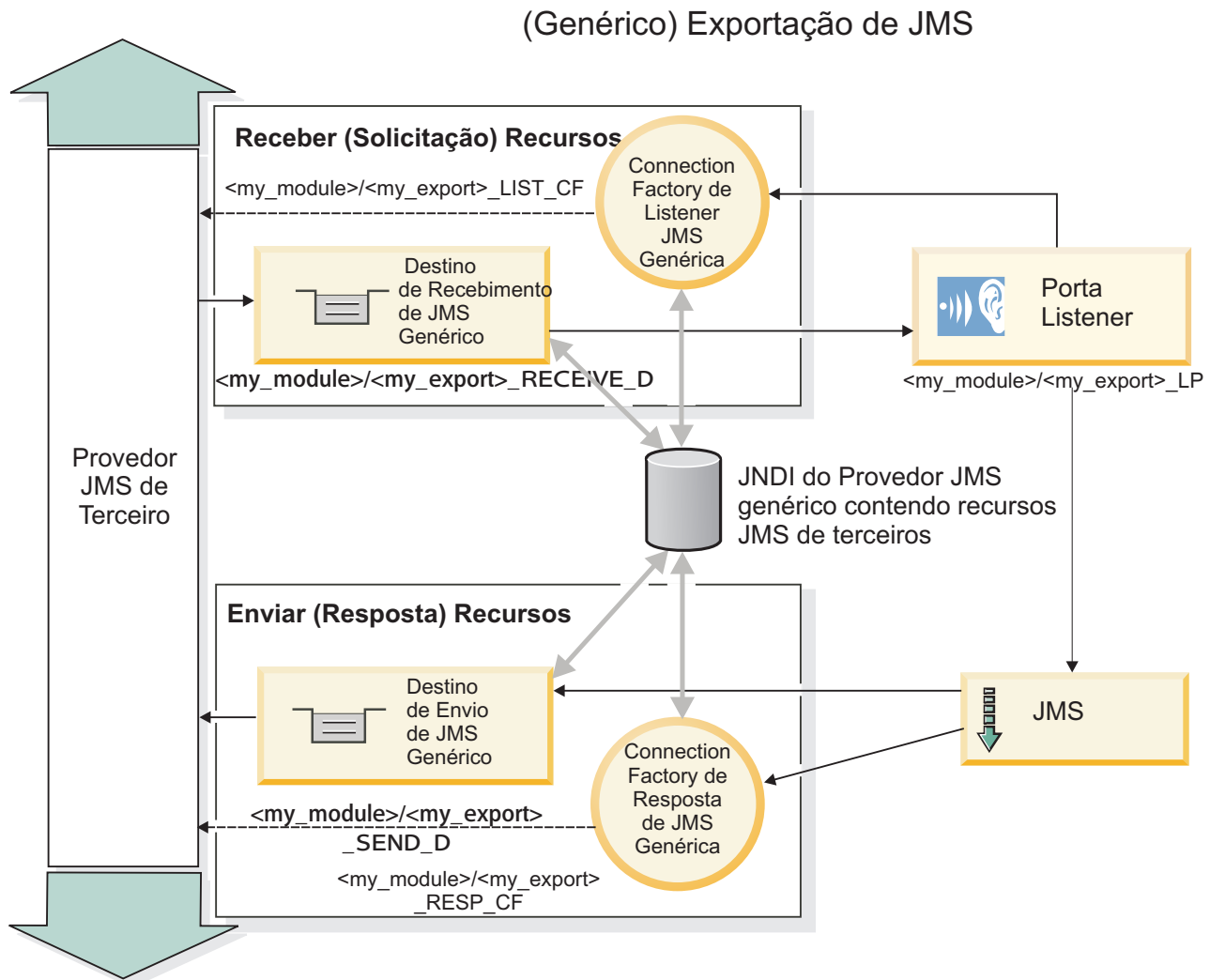


Figura 36. Recursos de Ligação de Exportação JMS Genérica

Recursos-Chave de Ligações JMS Genéricas:

Os recursos da ligação de importação e exportação JMS Genérica são consistentes com aqueles das ligações de importação JMS ou JMS do MQ integradas do WebSphere. Recursos-chave incluem definições de cabeçalho e o acesso a recursos Java EE existentes. No entanto, devido à sua natureza genérica, não há opções de conectividade específicas do provedor JMS e esta ligação tem capacidade limitada para gerar recursos na implementação e instalação.

Importações Genéricas

Como o aplicativo de importação JMS do MQ, a implementação JMS Genérica é assíncrona e suporta três chamadas: unidirecional, bidirecional (também conhecida como solicitação-resposta) e retorno de chamada.

Quando a importação JMS é implementada, um bean acionado por mensagens (MDB) fornecido pelo ambiente de tempo de execução é implementado. O MDB atende as respostas à mensagem de solicitação. O MDB está associado (atende no) ao destino enviado com a solicitação no campo de cabeçalho replyTo da mensagem JMS.

Exportações Genéricas

Ligações de exportação JMS genéricas diferem das ligações de exportação EIS em suas manipulações do retorno do resultado. Uma exportação JMS Genérica envia explicitamente a resposta para o destino replyTo especificado na mensagem recebida. Se nenhum for especificado, o envio de destino será usado.

Quando a exportação JMS Genérica é implementada, um bean acionado por mensagens (um MDB diferente daquele usado para importações JMS Genéricas) é implementado. Ele atende as solicitações recebidas no destino de recebimento e, em seguida, despacha as solicitações a serem processadas pelo tempo de execução SCA.

Cabeçalhos Especiais

Propriedades de cabeçalho especiais são usadas em importações e exportações JMS Genéricas para informar à ligação de destino como manipular a mensagem.

Por exemplo, a propriedade TargetFunctionName é usada pelo seletor de função padrão para identificar o nome da operação na interface de exportação que está sendo chamada.

Nota: A ligação de importação pode ser configurada para definir o cabeçalho TargetFunctionName para o nome da operação para cada operação.

Recursos Java EE

Vários recursos Java EE são criados quando uma ligação JMS é implementada em um ambiente Java EE.

- Porta listener para atender no destino de recebimento (resposta) (apenas bidirecional) para importações e no destino de recebimento (solicitação) para exportações
- Connection factory JMS genérica para outboundConnection (importação) e inboundConnection (exportação)
- Destino JMS genérico para os destinos de envio (importação) e de recebimento (exportação; apenas bidirecional)
- Connection factory JMS genérica para responseConnection (apenas bidirecional e opcional; caso contrário, outboundConnection será usada para importações e inboundConnection é usada para exportações)
- Destino JMS genérico para o destino de recebimento (importação) e de envio (exportação) (apenas bidirecional)
- Destino JMS de retorno de chamada do provedor de sistemas de mensagens padrão usado para acessar o destino de fila de retorno de chamada SIB (apenas bidirecional)
- Connection factory JMS de retorno de chamada de provedor de sistemas de mensagens padrão usada para acessar o destino JMS de retorno de chamada (apenas bidirecional)
- Destino de fila de retorno de chamada SIB usado para armazenar informações sobre a mensagem de solicitação para uso durante o processamento de resposta (apenas bidirecional)

A tarefa de instalação cria a ConnectionFactory, os três destinos e a ActivationSpec a partir das informações nos arquivos de importação e exportação.

Cabeçalhos JMS Genéricos:

Cabeçalhos JMS genéricos são Objetos de Dados de Serviço (SDO) que contêm todas as propriedades das propriedades de mensagem JMS Genéricas. Estas propriedades podem ser da mensagem de entrada ou elas podem ser as propriedades que serão aplicadas na mensagem de saída.

Uma mensagem JMS contém dois tipos de cabeçalhos - o cabeçalho do sistema JMS e diversas propriedades JMS. Ambos os tipos de cabeçalhos podem ser acessados em um módulo de mediação no Objeto de Mensagem de Serviço (SMO) ou usando a API ContextService.

As propriedades a seguir são configuradas estaticamente no methodBinding:

- JMSType
- JMSCorrelationID
- JMSDeliveryMode
- JMSPriority

A ligação JMS Genérica também suporta a modificação dinâmica de cabeçalhos e propriedades JMS da mesma maneira que as ligações JMS e JMS do MQ.

Alguns provedores JMS Genéricos impõem restrições sobre quais propriedades podem ser configuradas pelo aplicativo e em quais combinações. Você deve consultar a documentação do seu produto de terceiros para obter informações adicionais. No entanto, uma propriedade adicional foi incluída no methodBinding, ignoreInvalidOutboundJMSProperties, que permite que quaisquer exceções sejam propagadas.

Os cabeçalhos JMS Genéricos JMS e propriedades de mensagens são usados somente quando o comutador de ligação SCDL da arquitetura do componente de serviço de base está ativado. Quando o comutador está desativado, as informações de contexto são propagadas. Por padrão, esse comutador está ativado. Para evitar a propagação de informações de contexto, altere o valor para **false**.

Quando a propagação de contexto está ativada, as informações de cabeçalho têm permissão para fluir para a mensagem ou para o componente de destino. Para ativar e desativar a propagação de contexto, especifique **true** ou **false** para o atributo contextPropagationEnabled das ligações de importação e de exportação. Por exemplo:

```
<esbBinding xsi:type="eis:JMSImportBinding" contextProagationEnabled="true">
```

O padrão é **true**.

Resolução de Problemas de Ligações JMS Genéricas:

É possível diagnosticar e corrigir problemas com Ligação JMS Genérica.

Exceções de Implementação

Em resposta a várias condições de erro, a implementação de importação e exportação JMS Genérica pode retornar um dos dois tipos de exceções:

- Exceção de Negócios de Serviço: esta exceção é retornada se a falha especificada na interface de negócios de serviço (tipo de porta WSDL) ocorreu.
- Exceção de Tempo de Execução de Serviço: levantada em todos os outros casos. Na maioria dos casos, a exceção de causa conterà a exceção original (JMSException).

Resolução de Problemas de Expiração da Mensagem JMS Genérica

Uma mensagem de pedido pelo provedor JMS está sujeita à expiração.

Expiração de pedido refere-se à expiração de uma mensagem de pedido pelo provedor JMS quando o tempo da JMSExpiration na mensagem de pedido é atingido. Como com outras ligações JMS, a ligação JMS Genérica manipula a expiração da solicitação configurando a expiração na mensagem de retorno de chamada posicionada pela importação para ser igual para a solicitação realizada. A notificação da expiração da mensagem de retorno de chamada indicará que a mensagem de pedido expirou e que o cliente deve ser notificado por meio de uma exceção de negócios.

Se o destino do retorno de chamada for movido para um provedor terceirizado, no entanto, esse tipo de expiração de pedido não será suportado.

Expiração de resposta refere-se à expiração de uma mensagem de resposta pelo provedor JMS quando o tempo da JMSExpiration na mensagem de resposta é atingido.

A expiração de resposta para a ligação JMS genérica não é suportada, pois o comportamento de expiração exato de um provedor JMS terceirizado não é definido. Entretanto, você pode verificar se a resposta não expirou se e quando ela for recebida.

Para mensagens de pedido de saída, o valor JMSExpiration será calculado a partir do tempo aguardado e a partir dos valores requestExpiration contidos em asyncHeader, se configurados.

Resolução de Problemas de Erros da Connection Factory JMS Genérica

Quando você definir certos tipos de connection factories em seu provedor JMS Genérico, poderá receber uma mensagem de erro quando tentar iniciar um aplicativo. É possível modificar a connection factory externa para evitar este problema.

Ao ativar um aplicativo, você poderá receber a seguinte mensagem de erro:

```
MDB Listener Port JMSConnectionFactory type does not match
JMSDestination type
```

Esse problema pode surgir quando você estiver definindo connection factories externos. Especificamente, a exceção pode ser emitida quando você cria uma Factory de Conexão de Tópico do JMS 1.0.2, em vez de uma Connection Factory do JMS 1.1 (unificado) (ou seja, uma que seja capaz de suportar a comunicação ponto a ponto e de publicação/assinatura).

Para resolver esse problema, execute as seguintes etapas:

1. Acesse o provedor JMS Genérico que você está usando.
2. Substitua o JMS 1.0.2 Topic Connection Factory definido com um JMS 1.1 (unificado) Connection Factory.

Quando ativar o aplicativo com a Connection Factory JMS 1.1 recém-definida, você não deverá mais receber uma mensagem de erro.

Mensagens SCA Baseadas em JMS Genérico Não Estão Aparecendo no Gerenciador de Eventos com Falha

Se mensagens SCA originadas por meio de uma interação do JMS genérico falharem, você poderá encontrar essas mensagens no gerenciador de eventos com falha. Se tais mensagens não aparecerem no gerenciador de eventos com falha, certifique-se de que o valor da propriedade de máximo de novas tentativas na porta listener subjacente seja igual ou maior que 1. A configuração desse valor como 1 ou mais permite a interação com o gerenciador de eventos com falha durante chamadas de SCA para as ligações JMS genéricas.

Exceções de Manipulação:

A maneira na qual a ligação é configurada determina como as exceções que são levantadas por manipuladores de dados ou ligações de dados são manipuladas. Além disso, a natureza do fluxo de mediação determina o comportamento do sistema quando uma exceção desse tipo é lançada.

Uma variedade de problemas pode ocorrer quando um manipulador de dados ou uma ligação de dados é chamada por sua ligação. Por exemplo, um manipulador de dados pode receber uma mensagem que possui uma carga útil corrompida ou pode tentar ler uma mensagem que possua um formato incorreto.

A maneira que sua ligação trata uma exceção desse tipo é determinada por como você implementa o manipulador de dados ou a ligação de dados. O comportamento recomendado é que você projete sua ligação de dados para lançar uma **DataBindingException**.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, qualquer exceção do manipulador de dados é agrupada em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Quando qualquer exceção de tempo de execução, incluindo uma exceção **DataBindingException**, é lançada:

- Se o fluxo de mediação for configurado para ser transacional, a mensagem JMS será armazenada no Gerenciador de Evento com Falha por padrão para exclusão ou reprodução manual.

Nota: É possível alterar o modo de recuperação na ligação de modo que a mensagem seja retrocedida ao invés de ser armazenada no gerenciador de evento com falha.

- Se o fluxo de mediação não for transacional, a exceção será registrada e a mensagem será perdida.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, uma exceção do manipulador de dados é produzida em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Ligações JMS do WebSphere MQ:

A ligação JMS do WebSphere MQ fornece integração com aplicativos externos que usam um provedor baseado em JMS do WebSphere MQ.

Use as ligações de exportação e importação de JMS do WebSphere MQ para integrar diretamente com sistemas JMS ou JMS do MQ externos a partir de seu ambiente do servidor. Isso elimina a necessidade de usar os recursos Link do MQ ou Link do Cliente do Barramento de Integração de Serviços.

Quando um componente interage com um serviço baseado em JMS do WebSphere MQ por meio de uma importação, a ligação de importação de JMS do WebSphere MQ utiliza um destino para o qual os dados serão enviados e um destino no qual a resposta pode ser recebida. A conversão dos dados para e a partir de uma mensagem JMS é realizada por meio do componente de extremidade de ligação do manipulador de dados ou dos dados JMS.

Quando um módulo SCA fornece um serviço para clientes JMS do WebSphere MQ, a ligação de exportação de JMS do WebSphere MQ utiliza um destino no qual o pedido pode ser recebido e a resposta pode ser enviada. A conversão dos dados para e a partir de uma mensagem JMS é feita através do manipulador de dados ou da ligação de dados JMS.

O seletor de função fornece um mapeamento para a operação no componente de destino a ser chamado.

Visão Geral de Ligações JMS do WebSphere MQ:

A ligação JMS do WebSphere MQ fornece integração com aplicativos externos que usam o provedor JMS do WebSphere MQ.

Tarefas Administrativas do WebSphere MQ

Espera-se que o administrador do sistema do WebSphere MQ crie o Gerenciador de Filas do WebSphere MQ subjacente, o qual as ligações JMS do WebSphere MQ usarão, antes de executar um aplicativo contendo estas ligações.

Ligações de Importação JMS do WebSphere MQ

A importação JMS do WebSphere MQ permite que os componentes em seu módulo SCA se comuniquem com serviços fornecidos por provedores baseados em JMS do WebSphere MQ. Você deve estar utilizando uma versão suportada do WebSphere MQ. Requisitos detalhados de hardware e software podem ser localizados nas páginas de suporte IBM.

Dois tipos de cenários de uso para ligações de importação JMS do WebSphere MQ são suportados, dependendo do tipo de operação que está sendo chamado:

- **Unidirecional:** A importação JMS do WebSphere MQ coloca uma mensagem no destino de envio configurado na ligação de importação. Nada é enviado para o campo replyTo do cabeçalho JMS.
- **Bidirecional (solicitação-resposta):** A importação JMS do WebSphere MQ coloca uma mensagem no destino de envio.

O destino de recebimento é configurado no campo de cabeçalho replyTo. Um bean acionado por mensagens (MDB) é implementado para atender no destino de recebimento e, quando uma resposta é recebida, o MDB transmite a resposta de volta para o componente.

A ligação de importação pode ser configurada (utilizando o campo **Esquema de Correlação de Resposta** no Integration Designer) para esperar que o ID de correlação da mensagem de resposta tenha sido copiado do ID de mensagem de solicitação (o padrão) ou a partir do ID de correlação da mensagem de solicitação.

Para os cenários de uso unidirecional e bidirecional, propriedades do cabeçalho dinâmicas e estáticas podem ser especificadas. Propriedades estáticas podem ser configuradas a partir da ligação de método da importação JMS. Algumas destas propriedades tem significados especiais para o tempo de execução JMS SCA.

É importante observar que o JMS WebSphere MQ é uma ligação assíncrona. Se um componente de chamada chamar uma importação JMS do WebSphere MQ de maneira síncrona (para uma operação bidirecional), o componente de chamada será bloqueado até que a resposta seja retornada pelo serviço JMS.

Figura 37 na página 136 ilustra como a importação está vinculada ao serviço externo.

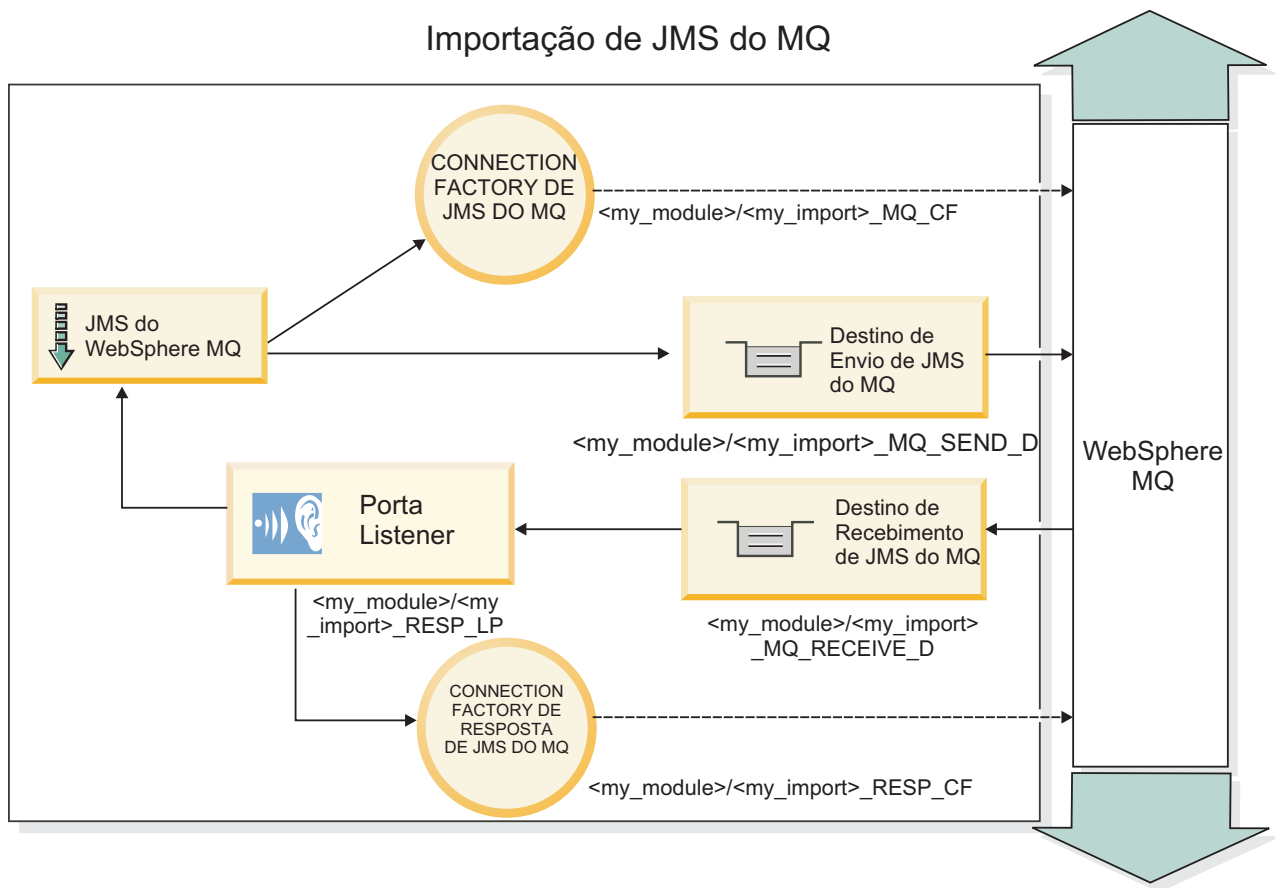


Figura 37. Recursos de Ligação da Importação JMS do WebSphere MQ

Ligações de Exportação JMS do WebSphere MQ

A ligação de exportação JMS do WebSphere MQ fornece os meios para que os módulos SCA forneçam serviços para aplicativos JMS externos no provedor JMS baseado no WebSphere MQ.

Um MDB é implementado para atender a solicitações recebidas para o destino de recebimento especificado na ligação de exportação. O destino especificado no campo Enviar é utilizado para enviar a resposta para a solicitação de entrada se o componente chamado fornecer uma resposta. O destino especificado no campo replyTo da mensagem de resposta substitui o destino especificado no campo Enviar.

O Figura 38 na página 137 ilustra como o solicitante externo está vinculado à exportação.

Exportação de JMS do MQ

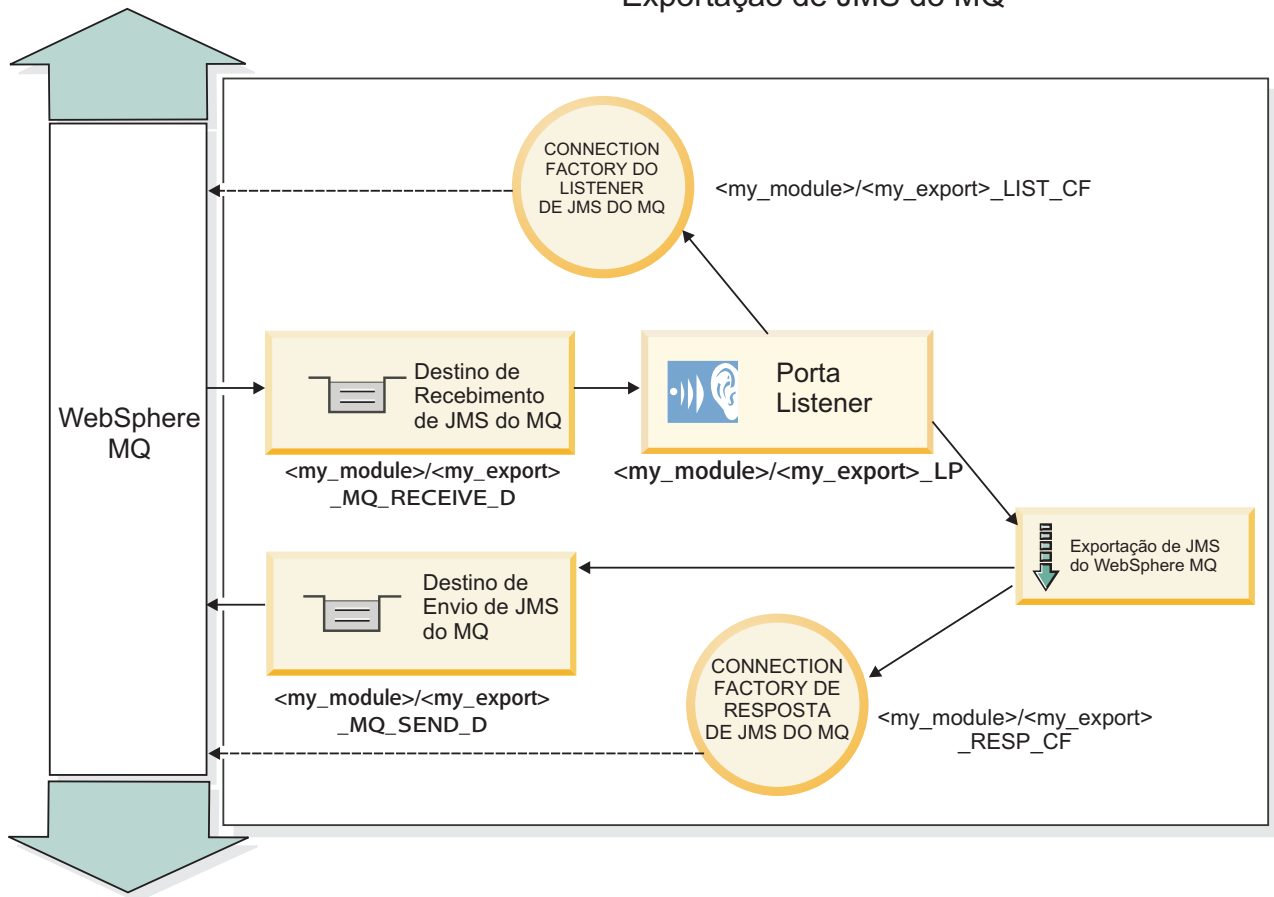


Figura 38. Recursos de Ligação de Exportação JMS do WebSphere MQ

Nota: Figura 37 na página 136 e Figura 38 ilustram como um aplicativo de uma versão anterior do IBM Business Process Manager está vinculado a um serviço externo. Para aplicativos desenvolvidos para IBM Business Process Manager Versão 7.0, a Especificação de ativação é usada ao invés da Porta Listener e da Connection Factory.

Recursos-chave das Ligações JMS do WebSphere MQ:

Os recursos-chave das ligações JMS do WebSphere MQ incluem cabeçalhos, artefatos Java EE e recursos Java EE criados.

Cabeçalhos

Um cabeçalho de mensagem JMS contém vários campos predefinidos que contêm valores usados por clientes e provedores para identificar e rotear mensagens. É possível usar propriedades de ligação para configurar esses cabeçalhos com valores fixos ou os cabeçalhos podem ser especificados dinamicamente no tempo de execução.

JMSCorrelationID

Links para uma mensagem relacionada. Geralmente, este campo é configurado com a sequência do identificador de mensagem da mensagem que está sendo respondida.

TargetFunctionName

Este cabeçalho é usado por um dos seletores de função fornecidos para identificar a operação que está sendo chamada. Configurar a propriedade de cabeçalho JMS TargetFunctionName em mensagens enviadas para uma exportação JMS permite que este seletor de função seja usado. A propriedade

pode ser configurada diretamente nos aplicativos clientes JMS ou ao conectar uma importação com uma ligação JMS tal como uma exportação. Neste caso, a ligação de importação JMS deve ser configurada para definir o cabeçalho TargetFunctionName para cada operação na interface com o nome da operação.

Esquemas de Correlação

As ligações JMS do WebSphere MQ fornecem vários esquemas de correlação que são usados para determinar como correlacionar mensagens de solicitação com mensagens de resposta.

RequestMsgIDToCorrelID

O JMSMessageID é copiado no campo JMSCorrelationID. Esta é a configuração padrão.

RequestCorrelIDToCorrelID

O JMSCorrelationID é copiado no campo JMSCorrelationID.

Recursos Java EE

Diversos recursos Java EE são criados quando uma importação JMS do MQ é implementada em um ambiente Java EE.

Parâmetros

Connection Factory do MQ

Usado pelos clientes para criar uma conexão com o provedor JMS do MQ.

Connection Factory de Resposta

Usado pelo tempo de execução JMS do MQ do SCA quando o destino de envio está em um Gerenciador de Filas diferente do destino de recebimento.

Especificação de Ativação

Uma especificação de ativação JMS do MQ está associada a um ou mais beans acionados por mensagens e fornece a configuração necessária para que eles recebam mensagens.

Destinos

- Destino de envio:
 - Importações: Onde a solicitação ou a mensagem de saída é enviada.
 - Exportações: Onde a mensagem de resposta será enviada se não for substituída pelo campo de cabeçalho JMSReplyTo da mensagem recebida.
- Destino de recebimento:
 - Importações: Onde a resposta ou a mensagem recebida deve ser colocada.
 - Exportações: Onde a mensagem recebida ou de solicitação deve ser colocada.

Cabeçalhos JMS:

Uma mensagem JMS contém dois tipos de cabeçalhos - o cabeçalho do sistema JMS e diversas propriedades JMS. Ambos os tipos de cabeçalhos podem ser acessados em um módulo de mediação no Objeto de Mensagem de Serviço (SMO) ou usando a API ContextService.

Cabeçalho do Sistema JMS

O cabeçalho do sistema JMS é representado no SMO pelo elemento JMSHeader, que contém todos os campos geralmente localizados em um cabeçalho JMS. Embora eles possam ser modificados na mediação (ou ContextService), alguns campos de cabeçalho do sistema JMS configurados no SMO não serão propagados na mensagem JMS de saída pois eles são substituídos pelo sistema ou por valores estáticos.

Os campos-chave no cabeçalho do sistema JMS que podem ser atualizados em uma mediação (ou ContextService) são:

- **JMSType** e **JMSCorrelationID** - valores das propriedades de cabeçalho de mensagem predefinidas específicas
- **JMSDeliveryMode** - valores para modo de entrega (persistente ou não persistente; o padrão é persistente)
- **JMSPriority** - valor de prioridade (0 a 9; padrão é JMS_Default_Priority)

Propriedades JMS

As propriedades JMS são representadas no SMO como entradas na lista de Propriedades. As propriedades podem ser incluídas, atualizadas ou excluídas em uma mediação ou usando a API ContextService.

As propriedades também podem ser configuradas estaticamente na ligação JMS. Propriedades que são configuradas estaticamente substituem as configurações (com o mesmo nome) que são configuradas dinamicamente.

As propriedades do usuário propagadas a partir de outras ligações (por exemplo, uma ligação HTTP) serão enviadas na ligação JMS como propriedades JMS.

Configurações de Propagação de Cabeçalho

A propagação do cabeçalho do sistema JMS e das propriedades a partir da mensagem JMS de entrada para componentes de recebimento de dados ou de componentes de envio de dados para a mensagem JMS de saída pode ser controlada pelo sinalizador Propagar Cabeçalho do Protocolo na ligação.

Quando Propagar Cabeçalho de Protocolo é configurado, as informações de cabeçalho têm permitidas para fluírem para a mensagem ou para o componente de destino, conforme descrito na lista a seguir:

- **Solicitação de exportação JMS**
O cabeçalho JMS recebido na mensagem será propagado para os componentes de destino por meio do serviço de contexto. As propriedades JMS recebidas na mensagem serão propagadas para componentes de destino por meio do serviço de contexto.
- **Resposta de exportação JMS**
Quaisquer campos de cabeçalho JMS configurados no serviço de contexto serão usados na mensagem de saída, se não substituídos por propriedades estáticas configuradas na ligação de exportação JMS. Quaisquer propriedades configuradas no serviço de contexto serão usadas na mensagem de saída se não substituídas por propriedades estáticas configuradas na ligação de exportação JMS.
- **Solicitação de importação JMS**
Quaisquer campos de cabeçalho JMS configurados no serviço de contexto serão usados na mensagem de saída, se não substituídos por propriedades estáticas configuradas na ligação de importação JMS. Quaisquer propriedades configuradas no serviço de contexto serão usadas na mensagem de saída se não substituídas por propriedades estáticas configuradas na ligação de importação JMS.
- **Resposta de importação JMS**
O cabeçalho JMS recebido na mensagem será propagado para os componentes de destino por meio do serviço de contexto. As propriedades JMS recebidas na mensagem serão propagadas para componentes de destino por meio do serviço de contexto.

Cientes Externos:

O servidor pode enviar mensagens para clientes externos ou receber mensagens deles usando ligações JMS do WebSphere MQ.

Um cliente externo (como um portal da web ou um Enterprise Information System) pode enviar uma mensagem para um componente SCA no aplicativo por meio de uma exportação ou pode ser chamado por um componente SCA no aplicativo por meio de uma importação.

A ligação de exportação JMS do WebSphere MQ implementa beans conduzidos por mensagens (MDBs) para atender as solicitações recebidas no destino de recebimento especificado na ligação de exportação. O destino especificado no campo de envio é usado para enviar a resposta para a solicitação de entrada se o aplicativo chamado fornece uma resposta. Assim, um cliente externo pode chamar aplicativos por meio da ligação de exportação.

As importações JMS do WebSphere MQ se ligam a clientes externos e podem entregar mensagem para eles. Esta mensagem pode ou não demandar uma resposta do cliente externo.

Mais informações sobre como interagir com clientes externos usando o WebSphere MQ podem ser localizadas no Centro de Informações do WebSphere MQ.

Resolução de Problemas de Ligações JMS do WebSphere MQ:

É possível diagnosticar e corrigir problemas com as ligações JMS do WebSphere MQ.

Exceções de Implementação

Em resposta a várias condições de erro, a implementação de importação e exportação de JMS do MQ pode retornar um de dois tipos de exceções:

- Exceção de Negócios de Serviço: esta exceção é retornada se a falha especificada na interface de negócios de serviço (tipo de porta WSDL) ocorreu.
- Exceção de Tempo de Execução de Serviço: levantada em todos os outros casos. Na maioria dos casos, a exceção de causa conterá a exceção original (JMSEException).

Por exemplo, uma importação espera apenas uma mensagem de resposta para cada mensagem de solicitação. Se mais de uma resposta chegar, ou se uma resposta atrasada (uma para a qual a expiração de resposta de SCA expirou) chegar, uma Exceção de Tempo de Execução de Serviço será emitida. A transação é retrocedida e a mensagem de resposta é retirada da fila ou manipulada pelo gerenciador de eventos com falha.

Mensagens SCA com Base no WebSphere MQ JMS Não Aparecem no Gerenciador de Eventos com Falha

Se mensagens SCA originadas por uma interação do WebSphere MQ JMS falharem, você pode encontrar essas mensagens no gerenciador de eventos com falha. Se tais mensagens não estiverem aparecendo no gerenciador de eventos com falha, certifique-se de que o valor da propriedade de máximo de novas tentativas na porta listener subjacente seja igual ou maior que **1**. A configuração deste valor como **1** ou mais permite a interação com o gerenciador de eventos com falha durante chamadas de SCA para as ligações JMS do MQ.

Cenários de Uso Incorreto: Comparação com Ligações WebSphere MQ

A ligação WebSphere MQ JMS é projetada para interoperar com aplicativos JMS implementados no WebSphere MQ, que expõe mensagens de acordo com o modelo de mensagem JMS. A importação e a exportação do WebSphere MQ, no entanto, são projetadas principalmente para interoperar com aplicativos WebSphere MQ nativos e expor o conteúdo integral do corpo da mensagem do WebSphere MQ para mediações.

Os cenários a seguir devem ser construídos utilizando a ligação do WebSphere MQ JMS, não a ligação do WebSphere MQ:

- Chamar um MDB (bean acionado por mensagens) JMS a partir de um módulo SCA, no qual o MDB é implementado no provedor WebSphere MQ JMS. Use uma importação JMS do WebSphere MQ.
- Permitir que o módulo SCA seja chamado a partir de um servlet de componente Java EE ou EJB por meio de JMS. Use uma exportação JMS do WebSphere MQ.

- Mediar o conteúdo de uma JMS MapMessage, em trânsito através do WebSphere MQ. Use uma exportação e importação JMS do WebSphere MQ em conjunto com o manipulador de dados ou a ligação de dados apropriada.

Há situações em que a ligação WebSphere MQ e a ligação WebSphere MQ JMS talvez sejam esperadas para interoperação. Em particular, quando fizer uma ponte entre os aplicativos WebSphere MQ Java EE e não Java EE, use uma exportação do WebSphere MQ e importação JMS do WebSphere MQ (ou vice-versa) em conjunto com ligações de dados ou módulos de mediação apropriados (ou ambos).

Exceções de Manipulação:

A maneira na qual a ligação é configurada determina como as exceções que são levantadas por manipuladores de dados ou ligações de dados são manipuladas. Além disso, a natureza do fluxo de mediação determina o comportamento do sistema quando uma exceção desse tipo é lançada.

Uma variedade de problemas pode ocorrer quando um manipulador de dados ou uma ligação de dados é chamada por sua ligação. Por exemplo, um manipulador de dados pode receber uma mensagem que possui uma carga útil corrompida ou pode tentar ler uma mensagem que possua um formato incorreto.

A maneira que sua ligação trata uma exceção desse tipo é determinada por como você implementa o manipulador de dados ou a ligação de dados. O comportamento recomendado é que você projete sua ligação de dados para lançar uma **DataBindingException**.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, qualquer exceção do manipulador de dados é agrupada em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Quando qualquer exceção de tempo de execução, incluindo uma exceção **DataBindingException**, é lançada:

- Se o fluxo de mediação for configurado para ser transacional, a mensagem JMS será armazenada no Gerenciador de Evento com Falha por padrão para exclusão ou reprodução manual.

Nota: É possível alterar o modo de recuperação na ligação de modo que a mensagem seja retrocedida ao invés de ser armazenada no gerenciador de evento com falha.

- Se o fluxo de mediação não for transacional, a exceção será registrada e a mensagem será perdida.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, uma exceção do manipulador de dados é produzida em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Ligações WebSphere MQ:

A ligação do WebSphere MQ fornece conectividadeService Component Architecture (SCA) com os aplicativos do WebSphere MQ.

Use as ligações de importação e exportação do WebSphere MQ para integrar diretamente a um sistema baseado no WebSphere MQ a partir do ambiente do servidor. Isso elimina a necessidade de usar os recursos Link do MQ ou Link do Cliente do Barramento de Integração de Serviços.

Quando um componente interagir com um serviço do WebSphere MQ por meio de uma importação, a ligação de importação do WebSphere MQ usará uma fila para a qual os dados serão enviados e uma fila em que a resposta possa ser recebida.

Quando um módulo SCA fornecer um serviço para os clientes do WebSphere MQ, a ligação de exportação do WebSphere MQ usará uma fila em que a solicitação poderá ser recebida e a resposta poderá ser enviada. O seletor de função fornece um mapeamento para a operação no componente de destino a ser chamado.

A conversão dos dados de carga útil para/de uma mensagem do MQ é feita por meio do manipulador de dados do corpo do MQ ou da ligação de dados. A conversão dos dados do cabeçalho para/de uma mensagem do MQ é feita por meio da ligação de dados do cabeçalho do MQ.

Para obter informações sobre as versões do WebSphere MQ suportadas, consulte a página da web *Requisitos Detalhados do Sistema*.

Visão Geral de Ligações do WebSphere MQ:

A ligação do WebSphere MQ fornece integração com aplicativos baseados em MQ nativos.

Tarefas Administrativas do WebSphere MQ

Espera-se que o administrador do sistema do WebSphere MQ crie o WebSphere MQ Queue Manager subjacente, que as ligações do WebSphere MQ irão usar, antes de executar um aplicativo contendo estas ligações.

Tarefas Administrativas do WebSphere

Você deve configurar a propriedade **Caminho da biblioteca nativa** do adaptador de recursos do MQ no Websphere para a versão do WebSphere MQ suportada pelo servidor e reiniciar o servidor. Isto garante que as bibliotecas de uma versão suportada do WebSphere MQ estejam sendo usadas. Os requisitos de hardware e software detalhados podem ser localizados nas páginas de suporte IBM.

Ligações de Importação do WebSphere MQ

A ligação de importação do WebSphere MQ permite que componentes em seu módulo SCA se comuniquem com serviços fornecidos por aplicativos externos baseados em WebSphere MQ. Você deve estar utilizando uma versão suportada do WebSphere MQ. Os requisitos de hardware e software detalhados podem ser localizados nas páginas de suporte IBM.

A interação com sistemas WebSphere MQ externos inclui o uso de filas para o envio de solicitações e recebimento de respostas.

Dois tipos de cenários de uso para a ligação de importação do WebSphere MQ são suportados, dependendo do tipo de operação que está sendo chamado:

- Unidirecional: A importação do WebSphere MQ coloca uma mensagem na fila configurada no campo **Fila de Destino de Envio** da ligação de importação. Nada é enviado para o campo replyTo do cabeçalho MQMD.
- Bidirecional (solicitação-resposta): A importação do WebSphere MQ coloca uma mensagem na fila configurada no campo **Fila de Destino de Envio**

A fila de recebimento é configurada no campo replyTo do cabeçalho MQMD. Um bean acionado por mensagens (MDB) é implementado para atender na fila de recebimento, e quando uma resposta é recebida, o MDB passa a resposta de volta para o componente.

A ligação de importação pode ser configurada (usando o campo **Esquema de Correlação de Resposta**) para esperar que o ID de correlação da mensagem de resposta tenha sido copiado do ID de mensagem de solicitação (o padrão) ou do ID de correlação da mensagem de solicitação.

É importante observar que o WebSphere MQ é uma ligação assíncrona. Se um componente de chamada chamar uma importação do WebSphere MQ de maneira síncrona (para uma operação bidirecional), o componente de chamada será bloqueado até a resposta ser retornada pelo serviço do WebSphere MQ.

Figura 39 ilustra como a importação está vinculada ao serviço externo.

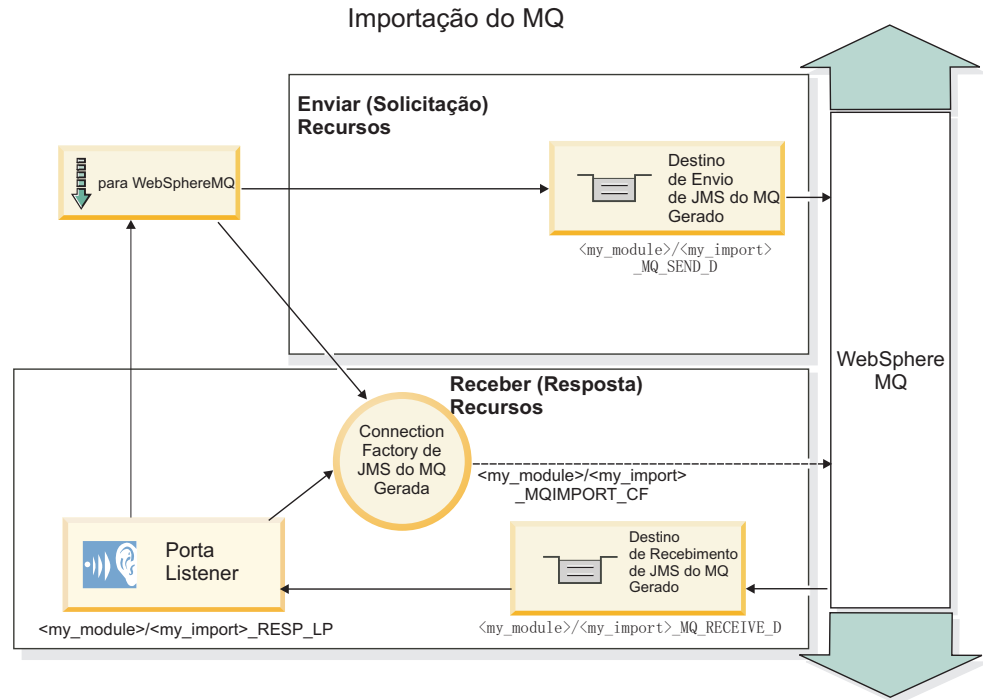


Figura 39. Recursos de Ligação de Importação do WebSphere MQ

Ligações de Exportação do WebSphere MQ

A ligação de exportação do WebSphere MQ fornece os meios para que os módulos SCA forneçam serviços para aplicativos externos baseados no WebSphere MQ.

Um MDB é implementado para atender a solicitações recebidas para a **Fila de destino de recebimento** especificada na ligação de exportação. A fila especificada no campo **Fila de Destino de Envio** é usada para enviar a resposta para a solicitação de entrada se o componente chamado fornecer uma resposta. A fila especificada no campo `replyTo` da mensagem de resposta substitui a fila especificada no campo **Fila de Destino de Envio**.

O Figura 40 na página 144 ilustra como o solicitante externo está vinculado à exportação.

Exportação do MQ

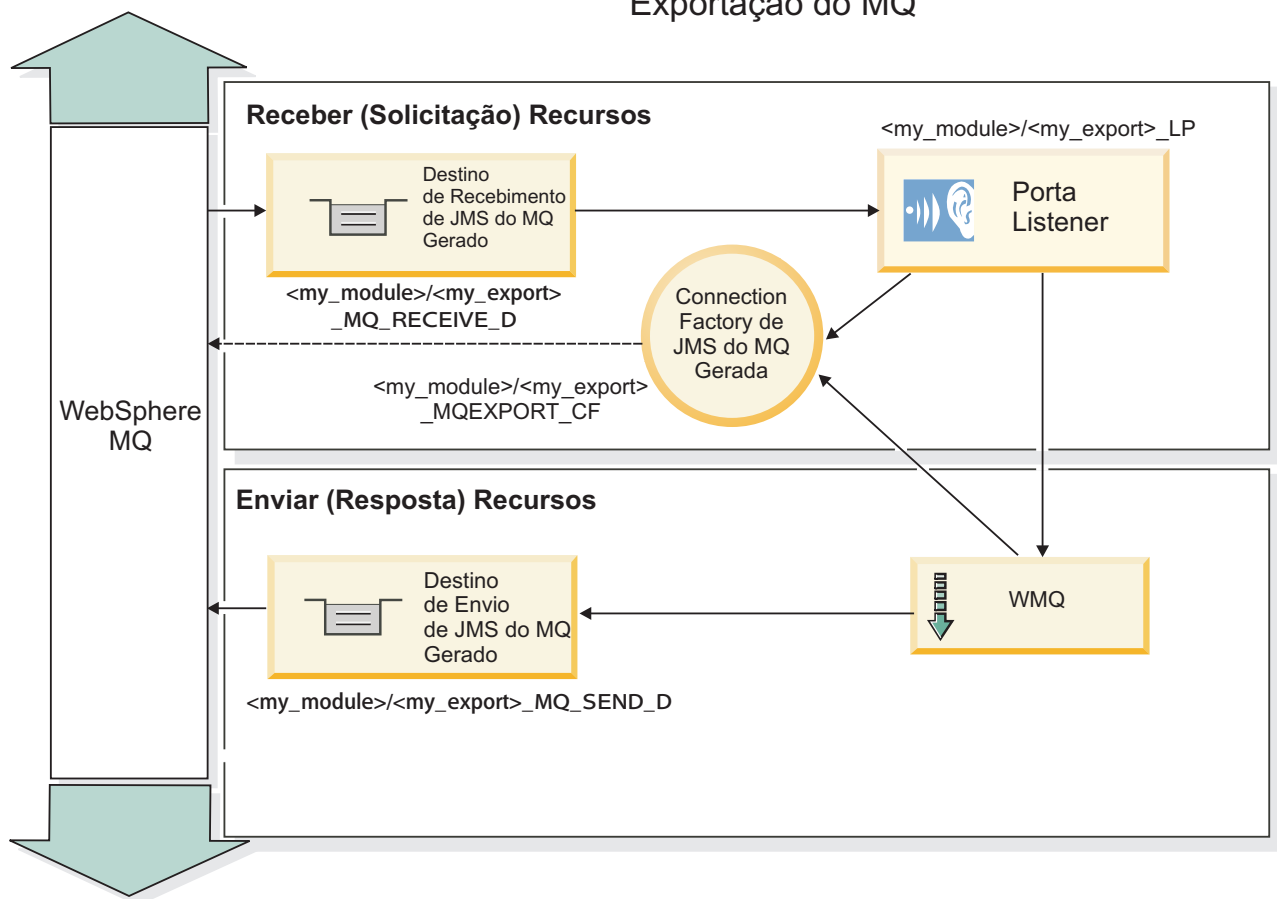


Figura 40. Recursos de Ligação de Exportação do WebSphere MQ

Nota: Figura 39 na página 143 e Figura 40 ilustram como um aplicativo de uma versão anterior do IBM Business Process Manager está vinculado a um serviço externo. Para aplicativos desenvolvidos para o IBM Business Process Manager Versão 7.x ou posterior, a especificação de ativação é usada, em vez da Porta do Listener e do Connection Factory.

Recursos-chave de uma Ligação do WebSphere MQ:

Os recursos-chave de uma ligação do WebSphere MQ incluem cabeçalhos, artefatos Java EE e recursos Java EE criados.

Esquemas de Correlação

Um aplicativo de solicitação/resposta do WebSphere MQ pode usar uma de várias técnicas para correlacionar as mensagens de resposta com solicitações, construídas em torno dos campos MQMD MessageID e CorrelID. Na grande maioria dos casos, o solicitante permite que o gerenciador de filas selecione um MessageID e espera que o aplicativo de resposta copie isso para o CorrelID da resposta. Na maioria dos casos, o solicitante e o aplicativo de resposta sabem, implicitamente, qual técnica de correlação está em uso. Ocasionalmente, o aplicativo de resposta honrará vários sinalizadores no campo Relatório da solicitação que descrevem como lidar com esses campos.

As ligações de exportação para o WebSphere MQ podem ser configuradas com as seguintes opções:

Opções MsgId de resposta:

Novo MsgID

Permite que o gerenciador de filas selecione um MsgId exclusivo para a resposta (padrão).

Copiar a partir do MsgID de Solicitação

Copia o campo MsgId a partir do campo MsgId na solicitação.

Copiar a partir da mensagem SCA

Configura o MsgId para que seja executado nos cabeçalhos do WebSphere MQ na mensagem de resposta SCA ou permite que o gerenciador de filas defina um novo ID se o valor não existir.

Como Opções de Relatório

Inspeciona o campo Relatório do MQMD na solicitação para obter uma dica de como manipular o MsgId. As opções MQRO_NEW_MSG_ID e MQRO_PASS_MSG_ID são suportadas e se comportam como Novo MsgId e Copiar de MsgID de Solicitação.

Opções de CorrelId de resposta:**Copiar a partir do MsgID de Solicitação**

Copia o campo CorrelId a partir do campo MsgId na solicitação (padrão).

Copiar a partir do CorrelID de Solicitação

Copia o campo CorrelId a partir do campo CorrelId na solicitação.

Copiar a partir da mensagem SCA

Configura o CorrelId para ser executado nos cabeçalhos do WebSphere MQ na mensagem de resposta SCA ou o deixa em branco se o valor não existir.

Como Opções de Relatório

Inspeciona o campo Relatório do MQMD na solicitação para obter uma dica de como manipular o CorrelId. As opções MQRO_COPY_MSG_ID_TO_CORREL_ID e MQRO_PASS_CORREL_ID são suportadas e se comportam como Copiar de MsgID de Solicitação e Copiar de CorrelID de Solicitação.

As ligações de importação para mensagens do WebSphere MQ podem ser configuradas com as opções a seguir:

Opções MsgId de solicitação:**Novo MsgID**

Permite que o gerenciador de filas selecione um MsgId exclusivo para a solicitação (padrão).

Copiar a partir da mensagem SCA

Configura o MsgId a ser executado nos cabeçalhos do WebSphere MQ na mensagem de solicitação SCA ou permite que o gerenciador de filas defina um novo Id se o valor não existir.

Opções de correlação de resposta:**Resposta possui CorrelID copiado de MsgId**

Espera que a mensagem de resposta tenha o campo CorrelId configurado, de acordo com o MsgId da solicitação (padrão).

Resposta possui MsgID copiado de MsgId

Espera que a mensagem de resposta tenha um campo MsgId configurado, de acordo com o MsgId da solicitação.

Resposta possui CorrelID copiado de CorrelId

Espera que a mensagem de resposta tenha um campo CorrelId configurado, de acordo com o CorrelId da solicitação.

Recursos Java EE

Vários recursos Java EE são criados quando uma ligação do WebSphere MQ é implementada em um ambiente Java EE.

Parâmetros

Connection Factory do MQ

Usado pelos clientes para criar uma conexão com o provedor do WebSphere MQ.

Connection Factory de Resposta

Usado pelo tempo de execução do MQ do SCA quando o destino de envio está em um Gerenciador de Filas diferente do destino de recebimento.

Especificação de Ativação

Uma especificação de ativação JMS do MQ está associada a um ou mais beans acionados por mensagens e fornece a configuração necessária para que eles recebam mensagens.

Destinos

- Destino de envio: onde a mensagem de solicitação ou de saída é enviada (importação); onde a mensagem de resposta será enviada (exportação), se não substituída pelo campo de cabeçalho do MQMD ReplyTo na mensagem recebida.
- Destino de recebimento: onde a mensagem de resposta/solicitação ou recebida deve ser colocada.

Cabeçalhos do WebSphere MQ:

Os cabeçalhos do WebSphere MQ incorporam determinadas convenções para conversão para as mensagens de Service Component Architecture (SCA).

As mensagens do WebSphere MQ consistem em um cabeçalho do sistema (o MQMD), zero ou mais outros cabeçalhos MQ (sistema ou customizado) e um corpo da mensagem. Se diversos cabeçalhos da mensagem existirem na mensagem, a ordem dos cabeçalhos será importante.

Cada cabeçalho contém informações que descrevem a estrutura do cabeçalho a seguir. O MQMD descreve o primeiro cabeçalho.

Como Cabeçalhos MQ são Analisados

Uma ligação de dados do Cabeçalho MQ é usada para analisar os cabeçalhos MQ. Os seguintes cabeçalhos são suportados automaticamente:

- MQRFH
- MQRFH2
- MQCIH
- MQIIH

Cabeçalhos que iniciam com **MQH** são manipulados de maneira diferente. Campos específicos do cabeçalho não são analisados; eles permanecem como bytes não analisados.

Para outros cabeçalhos do MQ, é possível gravar ligações de dados do cabeçalho MQ customizadas para analisar esses cabeçalhos.

Como Cabeçalhos MQ são Acessados

Os cabeçalhos MQ podem ser acessados no produto de uma de duas maneiras:

- Por meio do objeto de mensagem de serviço (SMO) em uma mediação
- Por meio da API ContextService

Cabeçalhos MQ são representados internamente com o elemento SMO MQHeader. MQHeader é um contêiner de dados de cabeçalho que estende MQControl, mas contém um elemento de valor de anyType. Ele contém o MQMD, o MQControl (informações de controle do corpo da mensagem MQ) e uma lista de outros cabeçalhos MQ.

- MQMD representa o conteúdo da descrição de mensagem do WebSphere MQ, exceto para informações que determinam a estrutura e a codificação do corpo.
- MQControl contém informações que determinam a estrutura e a codificação de um corpo da mensagem.
- MQHeaders contém uma lista de objetos MQHeader.

A cadeia do cabeçalho MQ é verificada para que, dentro do SMO, cada cabeçalho MQ transporte suas próprias informações de controle (CCSID, Codificação e Formato). Os cabeçalhos podem ser incluídos ou excluídos facilmente, sem alterar os outros dados do cabeçalho.

Configurando Campos no MQMD

É possível atualizar o MQMD usando a API de Contexto ou através do objeto de mensagem de serviço (SMO) em uma mediação. Os campos a seguir são propagados automaticamente para a mensagem do MQ de saída:

- Codificação
- Conjunto de Caracteres Codificados
- Formato
- Relatório
- Expiração
- Feedback
- Prioridade
- Persistência
- ID de Correlação
- Sinalizadores de Msg

Configure a ligação do MQ em uma Importação ou Exportação para propagar as propriedades a seguir para a mensagem MQ de saída:

MsgID

Configure **ID da Mensagem de Solicitação** como cópia da mensagem SCA.

MsgType

Limpe a caixa de seleção **Configurar tipo de mensagem como MQMT_DATAGRAM ou MQMT_REQUEST para operação de solicitação-resposta.**

ReplyToQ

Limpe a caixa de seleção **Substituir resposta na fila da mensagem de solicitação.**

ReplyToQMGr

Limpe a caixa de seleção **Substituir resposta na fila da mensagem de solicitação.**

Da versão 7.0 em diante, campos de contexto podem ser substituídos usando uma propriedade customizada na definição de destino de JNDI. Configure a propriedade customizada MDCTX com o valor SET_IDENTITY_CONTEXT no destino de envio para propagar os campos a seguir para a mensagem do MQ de saída:

- UserIdentifier
- AppIdentityData

Configure a propriedade customizada MDCTX com o valor SET_ALL_CONTEXT no destino de envio para propagar as propriedades a seguir para a mensagem do MQ de saída:

- UserIdentifier
- AppIdentityData
- PutApplType
- PutApplName
- ApplOriginData

Alguns campos não são propagados para a mensagem do MQ de saída. Os campos a seguir são substituídos durante o envio da mensagem:

- BackoutCount
- AccountingToken
- PutDate
- PutTime
- Deslocamento
- OriginalLength

Incluindo o MQCIH estaticamente em uma ligação do WebSphere MQ:

O IBM Business Process Manager suporta a inclusão de informações de cabeçalho MQCIH estaticamente sem usar um módulo de mediação.

Há várias maneiras de incluir as informações de cabeçalho MQCIH em uma mensagem (por exemplo, usando a primitiva de mediação do Configurador de Cabeçalho). Pode ser útil incluir estas informações de cabeçalho estaticamente, sem o uso de um módulo de mediação adicional. As informações de cabeçalho estático, incluindo o nome do programa CICS, o ID de transação e outros detalhes do cabeçalho de formato de dados, podem ser definidas e incluídas como parte da ligação do WebSphere MQ.

O WebSphere MQ, o MQ CICS Bridge e o CICS devem ser configurados para informações de cabeçalho MQCIH para serem incluídos estaticamente.

É possível usar Integration Designer para configurar a importação do WebSphere MQ com os valores estáticos que são necessários para as informações de cabeçalho MQCIH.

Quando uma mensagem chega e é processada pela importação do WebSphere MQ, é feita uma verificação para ver se as informações do cabeçalho MQCIH já estão presentes na mensagem. Se o MQCIH estiver presente, os valores estáticos definidos na importação do WebSphere MQ serão usados para substituir os valores dinâmicos correspondentes na mensagem. Se o MQCIH não estiver presente, um será criado na mensagem e os valores estáticos definidos na importação do WebSphere MQ serão incluídos.

Os valores estáticos definidos na importação do WebSphere MQ são específicos para um método. É possível especificar diferentes valores de MQCIH estáticos para métodos diferentes dentro da mesma importação do WebSphere MQ.

Este recurso não é usado para fornecer valores padrão se o MQCIH não contém informações do cabeçalho específicas porque um valor estático definido na importação do WebSphere MQ substituirá um valor correspondente fornecido na mensagem recebida.

Cientes Externos:

O IBM Business Process Manager pode enviar mensagens aos clientes externos e receber mensagens deles usando ligações do WebSphere MQ.

Um cliente externo (por exemplo, um portal da web ou um Enterprise Information System) pode enviar uma mensagem a um componente SCA no aplicativo por meio de uma exportação ou pode ser chamado por um componente SCA no aplicativo por meio de uma importação.

A ligação de exportação do WebSphere MQ implementa beans conduzidos por mensagens (MDBs) para atender as solicitações recebidas no destino de recebimento especificado na ligação de exportação. O destino especificado no campo de envio é usado para enviar a resposta para a solicitação de entrada se o aplicativo chamado fornece uma resposta. Assim, um cliente externo pode chamar aplicativos por meio da ligação de exportação.

As importações do WebSphere MQ se ligam aos clientes externos e podem entregar mensagem para eles. Esta mensagem pode ou não demandar uma resposta do cliente externo.

Mais informações sobre como interagir com clientes externos usando o WebSphere MQ podem ser localizadas no Centro de Informações do WebSphere MQ.

Resolução de Problemas de Ligações WebSphere MQ:

Você pode diagnosticar e corrigir falhas e condições de falhas que ocorrem com ligações WebSphere MQ.

Principais Condições de Falha

As principais condições de falha das ligações WebSphere MQ são determinadas pelas semântica transacional, pela configuração do WebSphere MQ ou pela referência ao comportamento existente em outros componentes. As principais condições de falha incluem:

- Falha de conexão com o gerenciador de fila ou a fila do WebSphere MQ.

Uma falha de conexão com o WebSphere MQ para receber mensagens resultará na Porta Listener MDB falhando ao iniciar. Essa condição será registrada no log do WebSphere Application Server. As mensagens persistentes permanecerão na fila do WebSphere MQ até que elas sejam recuperadas com sucesso (ou expiradas pelo WebSphere MQ).

Uma falha de conexão com o WebSphere MQ para enviar mensagens de saída causará o retrocesso da transação que controla o envio.

- Falha ao analisar uma mensagem de entrada ou ao construir uma mensagem de saída.

Uma falha na ligação de dados causa o retrocesso da transação que controla o trabalho.

- Falha ao enviar a mensagem de saída.

Uma falha ao enviar uma mensagem causa o retrocesso da transação relevante.

- Mensagens de resposta diversas ou inesperadas.

A importação espera apenas uma mensagem de resposta para cada mensagem de solicitação. Se mais de uma resposta chegar, ou se uma resposta atrasada (uma para a qual a expiração de resposta de SCA expirou) chegar, uma Exceção de Tempo de Execução de Serviço será emitida. A transação é retrocedida e a mensagem de resposta é retirada da fila ou manipulada pelo gerenciador de eventos com falha.

Cenários de Uso Incorreto: Comparação com Ligações WebSphere MQ JMS

A importação e exportação do WebSphere MQ são projetadas principalmente para interoperar com aplicativos nativos do WebSphere MQ e exportar o conteúdo completo do corpo da mensagem do

WebSphere MQ para mediações. A ligação WebSphere MQ JMS, entretanto, é projetada para interoperar com aplicativos JMS implementados no WebSphere MQ, que expõe mensagens de acordo com o modelo de mensagem JMS.

Os cenários a seguir devem ser construídos utilizando a ligação do WebSphere MQ JMS, não a ligação do WebSphere MQ:

- Chamar um MDB (bean acionado por mensagens) JMS a partir de um módulo SCA, no qual o MDB é implementado no provedor WebSphere MQ JMS. Use uma importação JMS do WebSphere MQ.
- Permitir que o módulo SCA seja chamado a partir de um servlet de componente Java EE ou EJB por meio de JMS. Use uma exportação JMS do WebSphere MQ.
- Mediar o conteúdo de uma JMS MapMessage, em trânsito através do WebSphere MQ. Use uma exportação e uma importação JMS do WebSphere MQ em conjunto com a ligação de dados apropriada.

Há situações em que a ligação WebSphere MQ e a ligação WebSphere MQ JMS talvez sejam esperadas para interoperação. Em particular, quando fizer uma ponte entre os aplicativos WebSphere MQ Java EE e não Java EE, use uma exportação do WebSphere MQ e importação JMS do WebSphere MQ (ou vice-versa) em conjunto com ligações de dados ou módulos de mediação apropriados (ou ambos).

Mensagens Não Entregues

Se o WebSphere MQ não puder entregar uma mensagem para seu destino desejado (devido a erros de configuração, por exemplo), ele enviará as mensagens para uma fila de devoluções nomeada.

Ao fazer isso, ele inclui um cabeçalho de devoluções no início do corpo da mensagem. Este cabeçalho contém os motivos da falha, o destino original e outras informações.

Mensagens SCA Baseadas em MQ Não Aparecerem no Gerenciador de Eventos com Falha

Se mensagens SCA se originarem devido a uma falha de interação do WebSphere MQ, você poderá encontrar essas mensagens no gerenciador de eventos com falha. Se estas mensagens não estão aparecendo no gerenciador de eventos com falha, verifique se o destino do WebSphere MQ subjacente tem um valor máximo de entregas com falha maior do que 1. A configuração deste valor como 2 ou mais permite a interação com o gerenciador de eventos com falha durante chamadas de SCA para ligações do WebSphere MQ.

Eventos com Falha do MQ são Reproduzidas no Gerenciador de Filas Errado

Quando uma connection factory predefinida deve ser usada para conexões de saída, as propriedades de conexão devem corresponder àquelas definidas na especificação de ativação usada para conexões de entrada.

A connection factory predefinida é usada para criar uma conexão ao reproduzir um evento com falha e deve, portanto, ser configurada para usar o mesmo gerenciador de filas a partir do qual a mensagem foi recebida originalmente.

Exceções de Manipulação:

A maneira na qual a ligação é configurada determina como as exceções que são levantadas por manipuladores de dados ou ligações de dados são manipuladas. Além disso, a natureza do fluxo de mediação determina o comportamento do sistema quando uma exceção desse tipo é lançada.

Uma variedade de problemas pode ocorrer quando um manipulador de dados ou uma ligação de dados é chamada por sua ligação. Por exemplo, um manipulador de dados pode receber uma mensagem que possui uma carga útil corrompida ou pode tentar ler uma mensagem que possua um formato incorreto.

A maneira que sua ligação trata uma exceção desse tipo é determinada por como você implementa o manipulador de dados ou a ligação de dados. O comportamento recomendado é que você projete sua ligação de dados para lançar uma **DataBindingException**.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, qualquer exceção do manipulador de dados é agrupada em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Quando qualquer exceção de tempo de execução, incluindo uma exceção **DataBindingException**, é lançada:

- Se o fluxo de mediação for configurado para ser transacional, a mensagem JMS será armazenada no Gerenciador de Evento com Falha por padrão para exclusão ou reprodução manual.

Nota: É possível alterar o modo de recuperação na ligação de modo que a mensagem seja retrocedida ao invés de ser armazenada no gerenciador de evento com falha.

- Se o fluxo de mediação não for transacional, a exceção será registrada e a mensagem será perdida.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, uma exceção do manipulador de dados é produzida em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Limitações de Ligações:

As ligações possuem algumas limitações em seus usos que estão listadas aqui.

Limitações da Ligação do MQ:

As ligações do MQ têm algumas limitações em seu uso que são listadas aqui.

Nenhuma Distribuição de Mensagem de Publicação-Assinatura

O método de distribuição de mensagens de publicação-assinatura atualmente não é suportado pela ligação do MQ, embora o WMQ em si suporte publicação-assinatura. No entanto, a ligação JMS do MQ suporta este método de distribuição.

Filas de Recebimento Compartilhadas

Diversas ligações de importação e exportação do WebSphere MQ esperam que quaisquer mensagens presentes em suas filas de recebimento configuradas sejam destinadas para essa exportação ou importação. Ligações de importação e exportação devem ser configuradas com as considerações a seguir:

- Cada importação do MQ deve ter uma fila de recebimento diferente porque a ligação de importação do MQ assume que todas as mensagens na fila de recebimento são respostas às solicitações que ela envia. Se a fila de recebimento for compartilhada por mais de uma importação, as respostas poderão ser recebidas pela importação incorreta e não serão correlacionadas com a mensagem de solicitação original.
- Cada exportação do MQ deve ter uma fila de recebimento diferente, porque, do contrário, você não pode prever qual exportação obterá qualquer mensagem de solicitação específica.
- Importações e exportações do MQ podem apontar para a mesma fila de envio.

Limitações das Ligações JMS, JMS do MQ e JMS Genéricas:

As ligações JMS e JMS do MQ possuem algumas limitações.

Implicações da Geração de Ligações Padrão

As limitações de uso das ligações JMS, JMS do MQ e JMS genéricas são discutidas nas seguintes seções:

- Implicações da Geração de Ligações Padrão
- Esquema de Correlação de Resposta
- Suporte Bidirecional

Quando você gerar uma ligação, vários campos serão preenchidos para você como os padrões, se não escolher inserir os valores você mesmo. Por exemplo, um nome de connection factory será criado para você. Se você souber que colocará seu aplicativo em um servidor e o acessará remotamente com um cliente, deverá, no momento da criação da ligação, inserir os nomes JNDI em vez de usar os padrões, pois você provavelmente deseja controlar esses valores através do console administrativo no tempo de execução.

No entanto, se você aceitou os padrões e descobriu posteriormente que não pode acessar seu aplicativo a partir de um cliente remoto, será possível usar o console administrativo para configurar explicitamente o valor da connection factory. Localize o campo de terminais do provedor nas configurações de connection factory e inclua um valor tal como <server_hostname>:7276 (se estiver usando o número da porta padrão).

Esquema de Correlação de Resposta

Se você usar o esquema de correlação de resposta CorrelationId To CorrelationId, usado para correlacionar mensagens em uma operação de solicitação-resposta, deverá ter um ID de correlação dinâmico na mensagem.

Para criar um ID de correlação dinâmico em um módulo de mediação usando o editor de fluxo de mediação, inclua uma primitiva de mediação Mapeamento antes da importação com a ligação JMS. Abra o editor de mapeamento. Os cabeçalhos de arquitetura de componente de serviço conhecidos estarão disponíveis na mensagem de destino. Arraste um campo que contém um ID exclusivo na mensagem de origem até o ID de correlação no cabeçalho JMS na mensagem de destino.

Suporte Bidirecional

Apenas caracteres ASCII são suportados para Java Naming and Directory Interface (JNDI) nomes no tempo de execução.

Filas de Recebimento Compartilhadas

Diversas Ligações de exportação e importação esperam que quaisquer mensagens presentes em suas filas de recebimento configuradas sejam destinadas para essa exportação ou importação. Ligações de importação e exportação devem ser configuradas com as considerações a seguir:

- Cada ligação de importação deve ter uma fila de recebimento diferente porque a ligação de importação assume que todas as mensagens na fila de recebimento são respostas para solicitações que são enviadas. Se a fila de recebimento for compartilhada por mais de uma importação, as respostas poderão ser recebidas pela importação incorreta e não serão correlacionadas com a mensagem de solicitação original.
- Cada exportação deve ter uma fila de recebimento diferente pois, caso contrário, você não poderá prever qual exportação obterá qualquer mensagem de solicitação específica.
- Importações e exportações podem apontar para a mesma fila de envio.

Objetos de Negócios

A indústria do software de computador desenvolveu vários modelos e estruturas de programação em que *objetos de negócios* fornecem uma representação natural dos dados de negócios para processamento do aplicativo.

Em geral, estes objetos de negócios:

- São definidos usando padrões de mercado
- Mapeiam dados de forma transparente para tabelas de banco de dados ou Enterprise Information Systems
- Suportam protocolos de chamada remota
- Fornecem a base do modelo de programação para programação de aplicativo

O Process Designer e o Integration Designer fornecem aos desenvolvedores um tipo de modelo de objeto de negócios comum para representar os diferentes tipos de entidades de negócios de diferentes domínios.

No Process Designer, o foco dos objetos de negócios é uma representação de tipo de dados. Os objetos de negócios de base (tipos de variáveis) são fornecidos nos kits de ferramentas do sistema e você pode criar tipos de variáveis customizados chamados objetos de negócios customizados. Consulte *Objetos de Negócios e Variáveis*.

No Integration Designer, que está disponível apenas com o IBM Business Process Manager Advanced, os objetos de negócios podem representar construções XSD mais complexas. No Integration Designer, os objetos de negócios possuem uma afinidade próxima com esquemas XML. Para obter informações sobre o que deve ser considerado ao integrar objetos de negócios definidos no Integration Designer a objetos de negócios definidos no Process Designer, consulte *Espelhamento da Biblioteca e Construções XML não Suportadas*.

No momento do desenvolvimento no Integration Designer, o modelo de objeto de negócios permite que os desenvolvedores definam objetos de negócios como definições de esquema XML. No tempo de execução, os dados de negócios definidos pelas definições do esquema XML são representados como *Objetos de Negócios Java*. Neste modelo, objetos de negócios são baseados livremente em rascunhos antigos da especificação de Objeto de Dados de Serviço (SDO) e fornecem o conjunto completo de interfaces de aplicativo do modelo de programação necessárias para manipular dados de negócios. Os subtópicos a seguir descrevem melhor como é possível usar objetos de negócios com o Integration Designer.

Definindo objetos de negócios

Você pode definir objetos de negócios usando o editor de objeto de negócios no Integration Designer. O editor de objeto de negócios armazena os objetos de negócios como definições de esquema XML.

O uso do esquema XML para definir objetos de negócios fornece várias vantagens:

- O esquema XML fornece um modelo de definição de dados baseado em padrões e uma base para interoperabilidade entre diferentes sistemas e aplicativos heterogêneos. O esquema XML é usado em conjunto com o Web Services Description Language (WSDL) para fornecer contratos de interface baseados em padrões entre componentes, aplicativos e sistemas.
- O esquema XML define um modelo de definição de dados rico para representar dados de negócios. Este modelo inclui tipos complexos, tipos simples, tipos definidos pelo usuário, herança de tipo e cardinalidade, entre outros recursos.
- Objetos de negócios podem ser definidos por interfaces de negócios e dados definidos no Web Services Description Language, bem como por esquema XML a partir de organizações padrões de mercado ou de outros sistemas e aplicativos. O Integration Designer pode importar esses objetos de negócios diretamente.

O Integration Designer também fornece suporte para descobrir os dados de negócios em bancos de dados e Enterprise Information Systems e, em seguida, gerar a definição de objeto de negócios de esquema XML baseado em padrões desses dados de negócios. Objetos de negócios gerados desta maneira são geralmente referidos como *objetos de negócios específicos de aplicativo* porque eles imitam a estrutura dos dados de negócios definidos no Enterprise Information System.

Quando um processo está manipulando dados de muitos sistemas de informações diferentes, pode ser útil transformar a representação diferente de dados de negócios (por exemplo, CustomerEIS1 e CustomerEIS2 ou OrderEIS1 e OrderEIS2) em uma representação canônica única (por exemplo, Cliente ou Pedido). A representação canônica é muitas vezes referida como o *objeto de negócios genérico*.

Definições de objeto de negócios, especialmente para objetos de negócios genéricos, são usadas com frequência por mais de um aplicativo. Para suportar essa reutilização, o Integration Designer permite que os objetos de negócios sejam criados nas bibliotecas que podem, então, ser associadas a diversos módulos de aplicativos.

O Web Services Description Language (WSDL) define os contratos para os serviços fornecidos e consumidos por um módulo aplicativo Service Component Architecture (SCA), bem como os contratos usados para criar os componentes dentro de um módulo aplicativo. Em um contrato, um WSDL pode representar as operações e os objetos de negócios (que são definidos pelo esquema XML para representar os dados de negócios).

Trabalhando com objetos de negócios

O Service Component Architecture (SCA) fornece a estrutura para definir um módulo aplicativo, os serviços que ele fornece, os serviços que ele consome e a composição de componentes que fornecem a lógica de negócios do módulo aplicativo. Objetos de negócios desempenham um papel importante no aplicativo, definindo os dados de negócios que são usados para descrever o serviço e contratos de componente e os dados de negócios que os componentes manipulam.

O diagrama a seguir descreve um módulo aplicativo SCA e ilustra muitos dos locais em que o desenvolvedor trabalha com objetos de negócios.

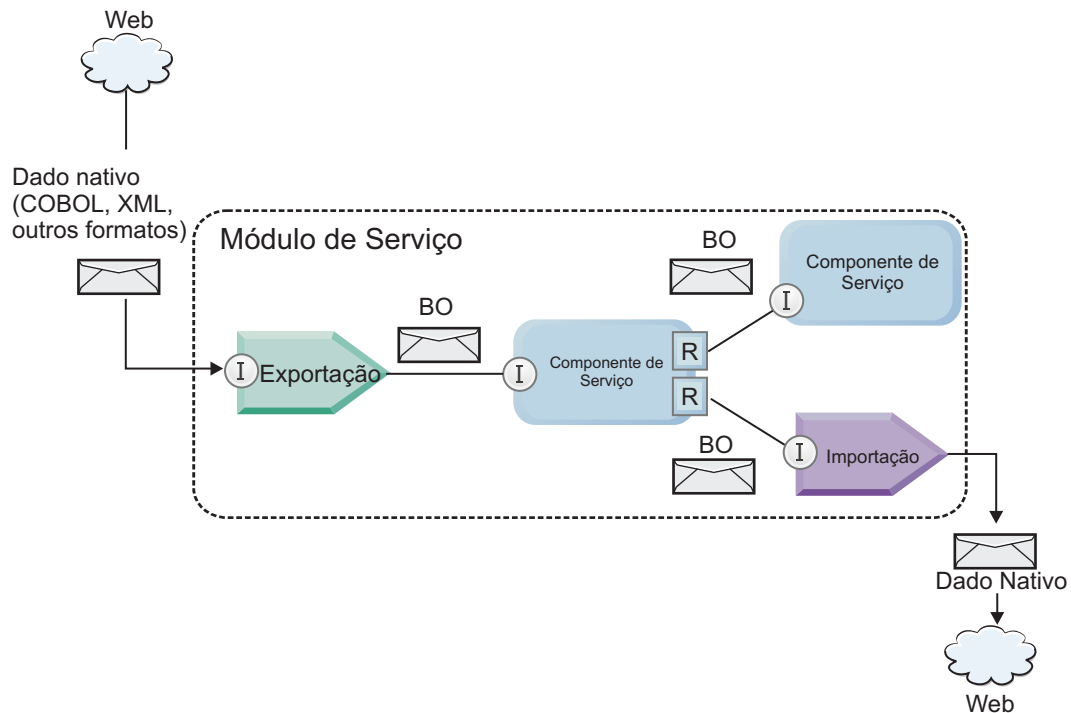


Figura 41. Objetos de Negócios Representam os Dados que Fluem entre Serviços em um Aplicativo

Nota: Este tópico descreve como os objetos de negócios são usados pelos módulos de aplicativo SCA. Se estiver usando interfaces Java, os módulos de aplicativo SCA também poderão processar objetos Java.

Modelo de Programação de Objeto de Negócios

O modelo de programação do objeto de negócios consiste em um conjunto de interfaces Java que representam:

- A definição do objeto de negócios e dados de instância
- Um conjunto de serviços que suportam as operações nos objetos de negócios

As definições de tipo de objeto de negócios são representadas pelas interfaces `commonj.sdo.Type` e `commonj.sdo.Property`. O modelo de programação de objeto de negócios fornece um conjunto de regras para mapear as informações de tipo complexo de esquema XML para a interface de Tipo e cada um dos elementos na definição de tipo complexo para a interface de Propriedade.

Instâncias de objetos de negócios são representadas pela interface `commonj.sdo.DataObject`. O modelo de programação do objeto de negócios é sem tipo, o que significa que a mesma interface `commonj.sdo.DataObject` pode ser usada para representar as definições de objeto de negócios diferentes, tais como Cliente e Pedido. A definição de quais propriedades podem ser configuradas e recuperadas a partir de cada objeto de negócios é determinada pelas informações de tipo definidas no esquema XML associado a cada objeto de negócios.

O comportamento do modelo de programação do objeto de negócios é baseado na especificação Service Data Object 2.1. Para obter informações adicionais, consulte a especificação SDO 2.1 para Java, tutoriais e javadocs na web: <http://www.oasis-opencsa.org/>.

Serviços de objeto de negócios suportam várias operações de ciclo de vida (como criação, igualdade, análise e serialização) nos objetos de negócios.

Para informações específicas sobre o modelo de programação do objeto de negócio, consulte Programando usando serviços de objeto de negócio e Documentação API e SPI gerada nos objetos de negócio.

Ligações, Ligações de Dados e Manipuladores de Dados

Conforme mostrado em Figura 41 na página 155, os dados de negócios que são usados para chamar serviços fornecidos pelos módulos de aplicativo SCA são transformados em objetos de negócios para que os componentes SCA possam manipular os dados de negócios. De modo semelhante, os objetos de negócios manipulados pelos componentes SCA são convertidos no formato de dados requerido pelos serviços externos.

Em alguns casos, tal como a ligação de serviço da Web, a ligação usada para exportar e importar serviços transforma automaticamente os dados no formato apropriado. Em outros casos, tal como a ligação JMS, os desenvolvedores podem fornecer uma ligação de dados ou o manipulador de dados que converte formatos não nativos em objetos de negócios representados pela interface DataObject.

Para obter informações adicionais sobre o desenvolvimento de ligações de dados e manipuladores de dados, consulte “Manipuladores de Dados” na página 59 e “Ligações de Dados” na página 61.

Componentes

Os componentes SCA definem seus contratos de serviço de provisão e consumo usando uma combinação do esquema Web Services Description Language e XML. Os dados de negócios que o SCA transmite entre componentes são representados como objetos de negócios usando a interface DataObject. O SCA verifica se esses tipos de objeto de negócios são compatíveis com o contrato de interface definido pelo componente a ser chamado.

As abstrações de modelo de programação para manipulação de objetos de negócios variam de componente para componente. O componente POJO e a primitiva Customizada do componente de fluxo de mediação fornecem a manipulação direta dos objetos de negócios ativando a programação Java usando diretamente as interfaces de programação e os serviços de objeto de negócios. A maioria dos componentes fornece abstrações de nível superior para a manipulação de objetos de negócios, mas também fornecem fragmentos de código Java para definir o comportamento customizado nas interfaces e serviços de objeto de negócios.

Os objetos de negócios podem ser transformados usando a combinação da Mediação do Fluxo de Interface e do componente Mapa de Objeto de Negócios ou do componente do fluxo de mediação e sua primitiva de Mapa XML. Esses recursos de transformação do objeto de negócios são úteis para converter objetos de negócios específicos do aplicativo para e a partir de objetos de negócios genéricos.

Objetos de Negócios Especiais

Os objetos de mensagem de serviço e gráficos de negócios são dois tipos especializados de objetos de negócios que são usados para propósitos de aplicativo específicos.

Objeto de mensagem de serviço

Um objeto de mensagem de serviço (SMO) é um objeto de negócios especializado que é usado pelos componentes de fluxo de mediação para representar a coleção de dados associados a uma chamada de serviço.

Um SMO possui uma estrutura fixa de nível superior que consiste em cabeçalhos, contexto, corpo e anexos (se presente).

- Os cabeçalhos transportam informações relacionadas à chamada de serviço sobre um protocolo ou uma ligação específica. Os exemplos são cabeçalhos SOAP e cabeçalhos JMS.

- Dados de contexto transportam informações lógicas adicionais associadas à chamada enquanto ela está sendo processada pelo componente do fluxo de mediação. Estas informações geralmente não fazem parte dos dados do aplicativo enviados ou recebidos por clientes.
- O corpo do SMO transporta os dados de negócios de carga útil, que representam os principais dados da mensagem do aplicativo ou da chamada na forma de um objeto de negócios padrão.

O SMO também pode transportar dados de anexo para chamadas de serviço da web usando SOAP com anexos.

Fluxos de mediação executam tais tarefas como roteamento de solicitação e transformação de dados e o SMO fornece a visualização combinada de conteúdo de cabeçalho e de carga útil em uma estrutura unificada única.

Gráfico de Negócios

Um gráfico de negócios é um objeto de negócios especial usado para fornecer suporte para sincronização de dados em cenários de integração.

Considere um exemplo no qual dois sistemas de informações corporativos possuem uma representação de um pedido específico. Quando o pedido é alterado em um sistema, uma mensagem pode ser enviada para o outro sistema para sincronizar os dados do pedido. Os gráficos de negócios suportam a noção de envio apenas na parte do pedido que foi alterada no outro sistema e anotando-a com informações de resumo da mudança para definir o tipo de alteração.

Neste exemplo, um gráfico de negócios Pedido seria transmitido para o outro sistema em que um dos itens de linha no pedido foi excluído e que a propriedade de data de envio projetada do pedido foi atualizada.

Os gráficos de negócios podem ser facilmente incluídos em objetos de negócios existentes no Integration Designer. Eles estão mais frequentemente localizados em cenários nos quais adaptadores do WebSphere estão sendo usados e para suportar a migração dos aplicativos do WebSphere InterChange Server.

Modo de Análise de Objeto de Negócios

O Integration Designer fornece uma propriedade em módulos e bibliotecas que você pode usar para configurar o modo de análise de XML para objetos de negócios para mais rápido ou lento.

- Se a opção for configurada como *mais rápido*, fluxos de bytes são XML serão analisados mais rápido para criar o objeto de negócios.
- Se a opção for configurada como *lento*, o objeto de negócios será criado normalmente, mas a análise real do fluxo de byte XML será adiada e parcialmente analisada apenas quando as propriedades do objeto de negócios forem acessadas.

No modo de análise XML, dados não XML são sempre analisados rapidamente para criar o objeto de negócios.

Considerações ao Escolher o Modo de Análise do Objeto de Negócios:

O modo de análise de objeto de negócios determina como os dados XML são analisados no tempo de execução. Um modo de análise de objeto de negócios é definido em um módulo ou uma biblioteca quando ela é criada. É possível alterar o modo de análise do módulo ou da biblioteca, no entanto, você deve estar ciente das implicações.

O modo de análise de objeto de negócios é configurado no nível do módulo e da biblioteca. Os módulos que foram criados em uma versão do IBM Integration Designer anterior à versão 7 serão executados no modo de análise mais rápido sem qualquer alteração necessária. Por padrão, os módulos e bibliotecas que são criados no IBM Integration Designer versão 7 e em versões mais recentes receberão o modo de

análise mais adequado dependendo de vários fatores, tal como o modo de análise de projetos existentes em sua área de trabalho ou o modo de análise de projetos dependentes ou outros projetos na mesma solução e assim por diante. É possível alterar o modo de análise do objeto de negócios de um módulo ou biblioteca para adequar sua implementação, no entanto, você deve estar ciente das considerações a seguir.

Considerações

- O modo de análise de objeto de negócios lento processa dados XML mais rápido; entretanto, há diferenças de compatibilidade entre o modo mais rápido e o modo lento que você precisa saber antes de alterar a configuração de um módulo ou uma biblioteca. Essas diferenças irão afetar o comportamento de tempo de execução dos módulos. Para obter informações sobre qual modo de análise é ideal para seu aplicativo, consulte "Benefícios do uso do modo de análise lenta versus rápida" nos links relacionados.
- Um módulo pode ser configurado apenas para executar em um modo de análise. As bibliotecas podem ser configuradas para suportar qualquer um dos modos de análise ou ambos os modos de análise. Uma biblioteca que é configurada para suportar ambos os modos de análise pode ser referenciada por um módulo usando o modo de análise mais rápido e um módulo usando o modo de análise lento. O modo de análise de uma biblioteca no tempo de execução é determinado pelos módulos que fazem referência à biblioteca. No tempo de execução, um módulo declara seu modo de análise e esse modo de análise é usado pelo módulo e quaisquer bibliotecas que o módulo usa.
- Módulos e bibliotecas que estão configurados para modos de análise diferentes são compatíveis nos seguintes casos:
 - Módulos e bibliotecas configurados com o modo de análise lento são compatíveis com bibliotecas que usam o modo de análise lento ou ambos os modos de análise, mais rápido e lento.
 - Módulos e bibliotecas configurados com o modo de análise mais rápido são compatíveis com bibliotecas que usam o modo de análise mais rápido ou ambos os modos de análise, mais rápido e lento.
 - Bibliotecas configuradas com os modos de análise lento e mais rápido são compatíveis apenas com bibliotecas que usam ambos os modos de análise, lento e mais rápido.
- Use o mesmo modo de análise para interagir os módulos que se comunicam usando a ligação SCA. Se os módulos se comunicarem usando modos de análise diferentes, isto poderá resultar em problemas de desempenho.

Conceitos relacionados:

"Benefícios de Usar o Modo de Análise Lento Versus Mais Rápido"

Alguns aplicativos se beneficiam do modo de análise de XML lento, enquanto outros veem o desempenho melhorado com modo de análise mais rápido. É recomendado que você teste o desempenho de seu aplicativo em ambos os modos de análise para determinar quais modo melhor se ajusta às características específicas de seu aplicativo.

Benefícios de Usar o Modo de Análise Lento Versus Mais Rápido:

Alguns aplicativos se beneficiam do modo de análise de XML lento, enquanto outros veem o desempenho melhorado com modo de análise mais rápido. É recomendado que você teste o desempenho de seu aplicativo em ambos os modos de análise para determinar quais modo melhor se ajusta às características específicas de seu aplicativo.

Aplicativos que analisam grandes fluxos de dados XML provavelmente verão as melhorias de desempenho quando o modo de análise XML lento é usado. Os benefícios de desempenho aumentam à medida que o tamanho do fluxo de byte XML aumenta e a quantidade de dados do fluxo de byte que é acessado pelo aplicativo diminui.

Os aplicativos a seguir provavelmente são executados melhor usando um modo de análise rápido:

- Aplicativos que analisam os fluxos de dados não XML
- Aplicativos que usam mensagens que são criadas usando o serviço BOFactory

- Aplicativos que analisam mensagens XML muito pequenas

Referências relacionadas:

“Considerações ao Escolher o Modo de Análise do Objeto de Negócios” na página 157

O modo de análise de objeto de negócios determina como os dados XML são analisados no tempo de execução. Um modo de análise de objeto de negócios é definido em um módulo ou uma biblioteca quando ela é criada. É possível alterar o modo de análise do módulo ou da biblioteca, no entanto, você deve estar ciente das implicações.

Considerações sobre Migração e Desenvolvimento de Aplicativo:

Se estiver configurando um aplicativo que foi desenvolvido originalmente usando o modo de análise mais rápido para usar agora o modo de análise lento, ou se estiver planejando para alternar um aplicativo entre o modo de análise lento ou mais rápido, esteja ciente das diferenças entre modos e as considerações ao alternar os modos.

Identificador de Erros

Se o fluxo de byte XML que está sendo analisado estiver malformatado, ocorrerão exceções de análise.

- No modo de análise de XML mais rápido, essas exceções ocorrem assim que o objeto de negócios é analisado a partir do fluxo de XML de entrada.
- Se o modo de análise de XML lento estiver configurado, as exceções de análise ocorrerão latentemente quando as propriedades do objeto de negócios forem acessadas e a parte do XML que está malformatada for analisada.

Para lidar com XML malformatado, selecione uma das opções a seguir:

- Implemente um barramento de serviço corporativo nas extremidades para validar o XML de entrada
- Crie a lógica de detecção de erro lenta no ponto em que as propriedades de objeto de negócios são acessadas

Pilhas e Mensagens de Exceções

Como os modos de análise de XML mais rápido e lento têm diferentes implementações subjacentes, os rastreios de pilha lançados pelas interfaces de programação e serviços de objeto de negócios possuem o mesmo nome da classe de exceção, mas eles podem não conter a mesma mensagem de exceção ou conjunto agrupado de classes de exceção específicas da implementação.

Formato de Serialização XML

O modo de análise de XML lento fornece uma otimização de desempenho que tenta copiar XML não modificado a partir do fluxo de bytes de entrada para o fluxo de bytes de saída na serialização. O resultado é o desempenho aprimorado, mas o formato de serialização do fluxo de byte XML de saída pode ser diferente se o objeto de negócios inteiro foi atualizado no modo de análise de XML lento ou se ele estava em execução no modo de análise de XML mais rápido.

Embora o formato de serialização de XML possa não ser precisamente sintaticamente equivalente, o valor semântico fornecido pelo objeto de negócios é equivalente independente dos modos de análise e XML pode ser seguramente transmitido entre os aplicativos em execução em diferentes modos de análise com equivalência de semântica.

Validador de Instância do Objeto de Negócios

O validador da instância do modo de objeto de negócios de análise de XML fornece uma validação com maior fidelidade de objetos de negócios, particularmente no aspecto da validação de valores da propriedade. Devido a essas melhorias, o validador de instância de modo de análise lento captura

problemas adicionais que não são capturados no modo de análise mais rápido e fornece mensagens de erro mais detalhadas.

Mapas XML Versão 602

Os fluxos de mediação originalmente desenvolvidos antes do WebSphere Integration Developer Versão 6.1 podem conter primitivas de Mapeamento que usam um mapa ou folha de estilo que não pode ser executados diretamente em modo de análise XML lento. Quando um aplicativo é migrado para uso em modo de análise de XML lento, os arquivos de mapa associados a primitivas de mapeamento podem ser automaticamente atualizados pelo assistente de migração para executar no novo modo. Entretanto, se uma primitiva Mapeamento referir-se diretamente a uma folha de estilo que foi editada manualmente, a folha de estilo não será migrada e não poderá ser executada em modo de análise XML lento.

APIs Não Publicadas Privadas

Se um aplicativo estiver tirando vantagem de interfaces de programação de objeto de negócios específicas da implementação, não publicadas, privadas, o aplicativo provavelmente falhará na compilação quando o modo de análise for alternado. No modo de análise mais rápido, essas interfaces privadas serão geralmente as classes de implementação de objeto de negócios definidas pelo Eclipse Modeling Framework (EMF).

Em todos os casos, é altamente recomendado que APIs privadas sejam removidas do aplicativo.

APIs do EMF do Objeto de Mensagem de Serviço

Um componente de mediação no IBM Integration Designer fornece a capacidade de manipular o conteúdo da mensagem usando as classes e interfaces Java fornecidas no pacote `com.ibm.websphere.sibx.smobo`. No modo de análise de XML lento, as interfaces Java no pacote `com.ibm.websphere.sibx.smobo` ainda podem ser usadas, mas os métodos que se referem diretamente às classes e interfaces do Eclipse Modeling Framework (EMF) ou que são herdados de interfaces do EMF provavelmente falharão.

O `ServiceMessageObject` e seu conteúdo não podem ser convertidos em objetos EMF no modo de análise de XML lento.

Serviço BOMode

O serviço `BOMode` é usado para determinar se o modo de análise de XML atualmente em execução é mais rápido ou lento.

Migração

Todos os aplicativos anteriores à versão 7.0.0.0 estão em execução no modo de análise mais rápido. Quando eles são migrados no tempo de execução usando as ferramentas de migração de tempo de execução BPM, eles continuam executando no modo de análise de XML mais rápido.

Para permitir que um aplicativo anterior à versão 7.0.0.0 seja configurado para usar o modo de análise de XML lento, primeiro você usa o Integration Designer para migrar os artefatos do aplicativo. Após a migração, você configura o aplicativo para usar a análise de XML lenta.

Consulte *Migrando Artefatos de Origem* para obter informações sobre como migrar artefatos no Integration Designer e consulte *Configurando o Modo de Análise de Objeto de Negócios de Módulos e Bibliotecas* para obter informações sobre como configurar o modo de análise.

Relacionamentos

Um relacionamento é uma associação entre duas ou mais entidades de dados, geralmente objetos de negócios. No IBM Business Process Manager Advanced, os relacionamentos podem ser usados para transformar dados equivalentes em objetos de negócios e outros dados, mas isso é representado de forma diferente ou pode ser usados para traçar associações por meio de diferentes objetos localizados em diferentes aplicativos. Eles podem ser compartilhados entre os aplicativos, soluções, e até mesmo em produtos.

O serviço de relacionamento no IBM Business Process Manager Advanced fornece a infra-estrutura e operações para gerenciar relacionamentos. Como ele permite que você lide com objetos de negócios independentemente de onde eles residem, ele pode fornecer uma visualização holística unificada em todos os aplicativos de uma empresa, e servem como um bloco de construção para soluções BPM. Como os relacionamentos são extensíveis e gerenciáveis, eles podem ser usados em soluções de integração complexas.

O Que São Relacionamentos?

Um relacionamento é uma associação entre objetos de negócios. Cada objeto de negócios em um relacionamento é chamado de *participante* no relacionamento. Cada participante no relacionamento é distinto dos outros participantes com base na função, ou *atribuição*, que ele serve nesse relacionamento. Um relacionamento contém uma lista de funções.

A *definição* de relacionamento descreve cada função e especifica como as funções são relacionadas. Também descreve o "formato" geral do relacionamento. Por exemplo, esta função pode ter apenas um participante, mas esta outra função pode tantos participantes quanto necessário. É possível definir um relacionamento *carro-proprietário*, por exemplo, onde um proprietário pode possuir diversos carros. Por exemplo, uma instância pode ter os seguintes participantes para cada uma destas funções:

- Carro (Ferrari)
- Proprietário (John)

A definição de relacionamento é um modelo para a *instância* de relacionamento. A instância é a instanciação do tempo de execução do relacionamento. No exemplo *car-owner*, uma instância pode descrever qualquer uma das seguintes associações:

- John possui Ferrari
- Sara possui Mazda
- Bob possui Ferrari

O uso de relacionamentos o libera da necessidade de construir persistência de rastreamento de relacionamento customizada dentro de sua lógica de negócios. Para determinados cenários, o serviço de relacionamento faz todo o trabalho para você. Consulte o exemplo descrito na seção em Relacionamentos de Identidades.

Cenários

A seguir há um exemplo típico de uma situação na qual uma solução de integração pode usar relacionamentos. Uma grande corporação compra diversas empresas, ou unidades de negócios. Cada unidade de negócios usa um software diferente para monitorar a equipe e notebooks. A empresa precisa de uma forma de monitorar seus funcionários e notebooks. Ela quer uma solução que permita:

- Visualizar todos os funcionários nas várias unidades de negócios como se estivessem em um banco de dados
- Ter uma única visualização de todos os seus notebooks
- Permitir que os funcionários efetuem login no sistema e comprem um notebook
- Acomodar os sistemas de aplicativos corporativos diferentes nas várias unidades de negócios

Para realizar isso, a empresa precisa de uma maneira de assegurar, por exemplo, que John Smith e John A. Smith em diferentes aplicativos sejam vistos como o mesmo funcionário. Por Exemplo, ela precisa de uma maneira de consolidar uma única entidade em diversos espaços do aplicativo.

Cenários de relacionamento mais complexos envolvem construir processos BPEL que desenham relacionamentos em diferentes objetos localizados em diversos aplicativos. Com cenários de relacionamento complexos, os objetos de negócios residem na solução de integração e não nos aplicativos. O serviço de relacionamento fornece uma plataforma para gerenciar relacionamentos de forma persistente. Antes do serviço de relacionamento, você precisaria construir seu próprio serviço de persistência de objeto. Dois exemplos de cenários de relacionamento complexos são:

- Você tem um objeto de negócios **carro** com um número VIN em um aplicativo SAP e deseja controlar o fato de que este carro é de propriedade de outra pessoa. No entanto, o relacionamento de propriedade está com alguém em um aplicativo PeopleSoft. Neste padrão de relacionamentos, você tem duas soluções e precisa construir uma ponte de ligação entre elas.
- Uma empresa de varejo grande deseja ser capaz de monitorar o merchandise retornado para dinheiro ou crédito. Há dois aplicativos diferentes envolvidos: um sistema de gerenciamento de pedidos (OMS) para compras e um sistema de gerenciamento devoluções (RMS) para devoluções. Os objetos de negócios residem em mais de um aplicativo e você precisa de uma maneira de mostrar os relacionamentos que existem entre eles.

Padrões de Uso Comuns

Os padrões de relacionamento mais comuns são padrões de *equivalência*. Eles são baseados em referência cruzada ou correlação. Há dois tipos de relacionamentos que se encaixam neste padrão: *sem identidade e identidade*.

- **Relacionamentos Sem Identidade** estabelecem associações entre objetos de negócios ou outros dados em uma base de um para muitos ou de muitos para muitos. Para cada instância de relacionamento, pode haver uma ou mais instâncias de cada participante. Um tipo de relacionamento de não identidade é um relacionamento de consulta estática. Um exemplo disso é um relacionamento no qual **CA** em um aplicativo SAP está relacionado com **California** em um aplicativo Siebel.

•

Relacionamentos de Identidades estabelecem associações entre objetos de negócios ou outros dados em uma base de um para um. Para cada instância de relacionamento, pode haver apenas uma instância de cada participante. Os relacionamentos de identidades capturam referências cruzadas entre objetos de negócios que são semanticamente equivalentes, mas que são identificados de maneira diferente em aplicativos diferentes. Cada participante do relacionamento está associado a um objeto de negócios que possui um valor (ou combinação de valores) que identifica exclusivamente o objeto. Relacionamentos de identidade geralmente transformam os atributos-chave de objetos de negócios, como números de ID e códigos do produto.

Por exemplo, se você tiver objetos de negócios **carro** nos aplicativos SAP, PeopleSoft e Siebel e desejar construir uma solução que os sincroniza, normalmente precisará introduzir a lógica de sincronização do relacionamento criada à mão em seis mapas:

SAP -> genérico

genérico -> SAP

PeopleSoft-> genérico

genérico-> PeopleSoft

Siebel-> Siebel

genérico-> Siebel

No entanto, se você usar relacionamentos em sua solução, o serviço de relacionamento fornecerá implementações padrão pré-construídas que mantêm todas essas instâncias de relacionamento para você.

Ferramentas para Trabalhar com Relacionamentos

O *editor de relacionamento* no Integration Designer é a ferramenta que você usa para modelar e projetar relacionamentos de integração de negócios e funções. Para obter informações complementares e de tarefas sobre a criação de relacionamentos e o uso do editor de relacionamentos, consulte Criando Relacionamentos.

O *serviço de relacionamento* é um serviço de infraestrutura no IBM Business Process Manager que mantém relacionamentos e funções no sistema e fornece operações para gerenciamento de relacionamentos e funções.

O *gerenciador de relacionamento* é a interface administrativa para gerenciar relacionamentos. Ele é acessado por meio das páginas do Gerenciador de Relacionamento do console administrativo.

Os relacionamentos podem ser chamados programaticamente por meio das APIs do serviço de relacionamento.

Serviço de Relacionamento

O serviço de relacionamento armazena dados do relacionamento em tabelas de relacionamentos, onde ele mantém o controle de valores específicos do aplicativo entre aplicativos e soluções. O serviço de relacionamento fornece operações para relacionamento e gerenciamento de função.

Como Funcionam os Relacionamentos

Relacionamentos e funções são definidos usando a interface gráfica da ferramenta editor de relacionamento no Integration Designer. O serviço de relacionamento armazena os dados de correlação em tabelas no banco de dados de relacionamento na origem de dados padrão que você especifica ao configurar o serviço de relacionamento. Uma tabela separada (às vezes chamada de tabela do participante) armazenar informações para cada participante no relacionamento. O serviço de relacionamento usa essas tabelas de relacionamentos para manter o controle dos valores específicos do aplicativo relacionados e propagar informações atualizadas em todas as soluções.

Relacionamentos, que são artefatos de negócios, são implementados em um projeto ou em uma biblioteca compartilhada. Na primeira implementação, o serviço de relacionamento preenche os dados.

No tempo de execução, quando mapas ou outros componentes do IBM Business Process Manager precisam de uma instância de relacionamento, as instâncias do relacionamento são atualizadas ou recuperadas, dependendo do cenário.

Dados da instância de relacionamento e função podem ser manipulados por três meios:

- Chamadas de fragmento Java do componente IBM Business Process Manager das APIs de serviço de relacionamento
- Transformações de relacionamento no serviço de mapeamento de objeto de negócios do IBM Business Process Manager
- A ferramenta do gerenciador de relacionamentos

Para obter informações complementares e de tarefas detalhadas sobre a criação de relacionamentos, a identificação de tipos de relacionamento e o uso do editor de relacionamentos, consulte o tópico Criando Relacionamentos.

Gerenciador de Relacionamento

O gerenciador de relacionamentos é a interface administrativa para gerenciar relacionamentos. Ele é acessado por meio das páginas do Gerenciador de Relacionamento do console administrativo.

O gerenciador de relacionamentos fornece uma interface gráfica com o usuário para criação e manipulação de dados de relacionamento e função no tempo de execução. É possível gerenciar entidades de relacionamento em todos os níveis: níveis da instância de relacionamento, da instância de função e de dados do atributo e dados de propriedade. Com o gerenciador de relacionamentos, é possível:

- Visualizar uma lista dos relacionamentos no sistema e informações detalhadas para relacionamentos individuais
- Gerenciar instâncias de relacionamento:
 - Consultar dados de relacionamento para visualizar subconjuntos de dados de instância
 - Consultar dados de relacionamento para visualizar subconjuntos de dados de instância usando visualizações do banco de dados
 - Visualizar uma lista de instâncias de relacionamento que correspondem a uma consulta de relacionamento e informações detalhadas sobre uma instância
 - Editar os valores da propriedade para uma instância de relacionamento
 - Criar e excluir instâncias de relacionamento
- Gerenciar funções e instâncias de função:
 - Visualizar detalhes sobre uma função ou uma instância de função
 - Editar propriedades da instância de função
 - Criar e excluir instâncias de função para um relacionamento
 - Retroceder dados da instância de relacionamento para um ponto no tempo quando você sabe que os dados são confiáveis
- Importar dados de um relacionamento estático existente em seu sistema ou exportar dados a partir de um relacionamento estático existente para um arquivo CSV ou RI
- Remover esquema de relacionamento e dados do repositório quando o aplicativo que os usa é desinstalado

Relacionamentos em Ambientes de Implementação de Rede

Os relacionamentos podem ser usados nos ambientes de Implementação de Rede (ND) sem nenhuma configuração extra.

Nos ambientes de Implementação de Rede (ND), os relacionamentos são instalados em um cluster de aplicativo. Relacionamentos ficam, então, visíveis dentro do cluster e todos os servidores no cluster têm acesso aos dados de instância armazenados no banco de dados de relacionamento. A capacidade de executar o serviço de relacionamento em um ambiente de ND o torna escalável e altamente disponível.

O gerenciador de relacionamentos permite que relacionamentos sejam gerenciados em clusters diferentes por meio de uma interface administrativa centralizada. Você conecta o gerenciador de relacionamentos a um servidor em um cluster selecionando seu MBean de relacionamento.

APIs do Serviço de Relacionamento

Os relacionamentos podem ser chamados programaticamente por meio das APIs do serviço de relacionamento, dentro ou fora de mapas de objetos de negócios.

Três tipos de API estão disponíveis:

- APIs de manipulação de instância de relacionamento (incluindo criar, atualizar, excluir dados de instância diretamente)
- APIs de suporte padrão de relacionamento (incluindo `correlate()`, `correlateforeignKeyLookup`)
- Padrões de consulta de relacionamento (APIs de consulta)

O Barramento de Serviço Corporativo no IBM Business Process Manager

O IBM Business Process Manager suporta a integração de serviços de aplicativo, incluindo os mesmos recursos que o WebSphere Enterprise Service Bus.

Conectando Serviços Através de um Enterprise Service Bus

Com um barramento de serviço corporativo (ESB), é possível maximizar a flexibilidade de um SOA. Participantes em uma interação de serviço são conectados ao ESB, em vez de diretamente um com o outro.

Quando o solicitante de serviços se conecta ao ESB, o ESB assume a responsabilidade pela entrega de suas solicitações, usando mensagens, para um provedor de serviços que oferece a função e qualidade de serviço necessárias. O ESB facilita as interações entre solicitante-provedor e trata protocolos, padrões de interação ou recursos de serviço incompatíveis. Um ESB também pode ativar ou aprimorar o monitoramento e o gerenciamento. O ESB fornece recursos de virtualização e gerenciamento que implementam e estendem os recursos principais de SOA.

O ESB abstrai os seguintes recursos:

Local e Identidade

Os participantes não têm que conhecer o local ou identidade dos outros participantes. Por exemplo, os solicitantes não têm que estar ciente de que uma solicitação possa ser atendida por algum dos vários provedores; os provedores de serviço podem ser incluídos ou removidos sem interrupção.

Protocolo de Interação

Os participantes não têm que compartilhar o mesmo protocolo de comunicação ou estilo de interação. Por exemplo, uma solicitação expressa como SOAP sobre HTTP pode ser verificada por um provedor que só entende SOAP sobre Sistema de Mensagens Java (JMS).

Interface

Os solicitantes e provedores não têm que concordar com uma interface comum. Um ESB reconcilia as diferenças transformando as mensagens de solicitação e de resposta em um formato esperado pelo provedor.

Qualidades de Serviço (Interação)

Os participantes, ou administradores do sistema, declaram seus requisitos de qualidade de serviço, incluindo autorização de solicitações, criptografia e descriptografia de conteúdo da mensagem, auditoria automática de interações de serviço e como suas solicitações devem ser roteadas (por exemplo, otimizando para velocidade ou custo).

Interposição do ESB entre os participantes permite que você ajuste suas interações por meio de uma construção lógica chamada de *mediação*. As mediações operam em mensagens em andamento entre solicitantes e provedores. Por exemplo, as mediações podem ser usadas para localizar serviços com características específicas que um solicitante está solicitando ou para resolver diferenças de interface entre solicitantes e provedores. Para interações complexas, as mediações podem ser encadeadas sequencialmente.

Usando mediações, um barramento de serviço corporativo realiza as seguintes ações entre o solicitante e o serviço:

- *Roteamento* de mensagens entre os serviços. Um barramento de serviço corporativo oferece uma infraestrutura de comunicação comum que pode ser usada para conectar serviços e, portanto, as funções de negócios que eles representam, sem a necessidade de os programadores gravarem e manterem uma lógica de conectividade complexa.
- *Conversão* de protocolos de transporte entre o solicitante e o serviço. Um barramento de serviço corporativo fornece uma maneira consistente, baseada em padrões, para integrar funções de negócios que usam diferentes padrões de TI. Isso permite a integração de funções de negócios que normalmente

não poderiam se comunicar, como conectar aplicativos em silos departamentais ou permitir que aplicativos em empresas diferentes participem em interações de serviços.

- *Transformação de formatos da mensagem* entre o solicitante e o serviço. O barramento de serviço corporativo permite que funções de negócios troquem informações em diferentes formatos, com o barramento assegurando que as informações entregues em uma função de negócios estejam no formato requerido por esse aplicativo.
- *Manipulação de eventos de negócio* a partir de origens diferentes. Um barramento de serviço corporativo suporta interações baseadas em eventos além das trocas de mensagens para manipular solicitações de serviço.

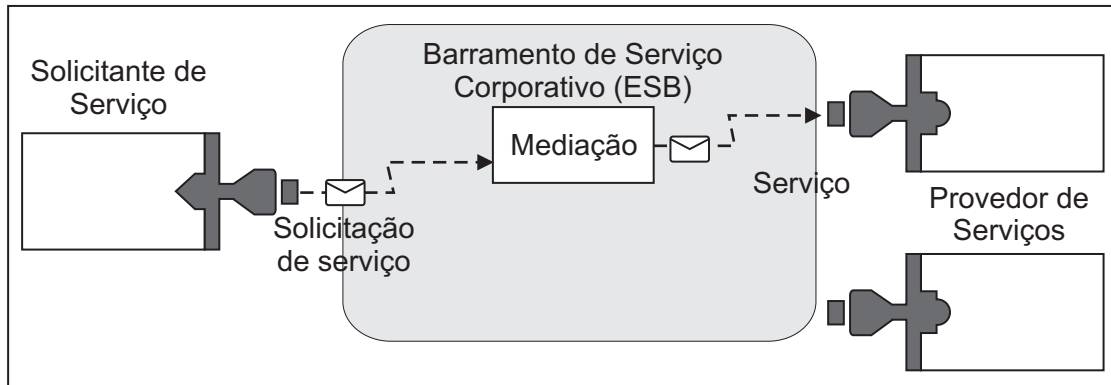


Figura 42. Um Barramento de Serviço Corporativo. O barramento de serviço corporativo está roteando as mensagens entre os aplicativos, os quais são solicitantes e provedores de serviços. O barramento está convertendo protocolos de transporte e transformando formatos da mensagem entre solicitantes e provedores. Nesta figura, cada aplicativo usa um protocolo diferente (representado pelos diferentes formatos geométricos de seus conectores) e usa diferentes formatos de mensagens.

Ao usar o barramento de serviço corporativo, é possível focar em seu negócio principal em vez de seus sistemas de computador. É possível alterar ou incluir nos serviços se necessário; por exemplo, para responder às mudanças da necessidade de negócios, para incluir capacidade extra de serviço ou incluir novos recursos. É possível fazer as alterações necessárias reconfigurando o barramento, com pouco ou nenhum impacto nos serviços e aplicativos existentes que usam o barramento.

Infraestrutura do Sistema de Mensagens do Barramento de Serviço Corporativo

O IBM Business Process Manager inclui recursos de barramento de serviço corporativo. O IBM Business Process Manager suporta a integração de tecnologias orientadas a serviço, orientadas a mensagem e conduzidas por evento para fornecer uma infraestrutura do sistema de mensagens baseada em padrões em um barramento de serviço corporativo integrado.

Os recursos de serviço corporativo que podem ser usados para seus aplicativos corporativos fornecem não apenas uma camada de transporte mas o suporte de mediação para facilitar as interações de serviço. O barramento de serviço corporativo é construído em torno de padrões abertos e arquitetura orientada a serviços (SOA). Ele é baseado na infraestrutura Java EE robusta e em serviços de plataforma associados fornecidos pelo IBM WebSphere Application Server Network Deployment.

O IBM Business Process Manager é desenvolvido com a mesma tecnologia disponível com o IBM WebSphere Enterprise Service Bus. Este recurso faz parte da funcionalidade subjacente do IBM Business Process Manager e nenhuma licença adicional para WebSphere Enterprise Service Bus é necessária para tirar vantagem destes recursos.

Entretanto, é possível implementar licenças independentes adicionais do WebSphere Enterprise Service Bus em sua empresa para estender o alcance da conectividade das soluções de integração de processos desenvolvidas com IBM Business Process Manager. Por exemplo, o WebSphere Enterprise Service Bus pode ser instalado mais próximo a um aplicativo SAP para hospedar um IBM WebSphere Adapter para

SAP e transformar mensagens SAP antes de enviar essas informações na rede para um processo de negócios coreografado pelo IBM Business Process Manager.

Hosts do Sistema de Mensagens ou do Destino de Fila:

Um host do sistema de mensagens ou do destino de fila fornece a função de sistema de mensagens dentro de um servidor. Um servidor torna-se o host de destino do sistema de mensagens quando você o configura como o destino do sistema de mensagens.

Um mecanismo do sistema de mensagens é executado em um servidor. O mecanismo do sistema de mensagens fornece funções do sistema de mensagens e um ponto de conexão para aplicativos se conectarem ao barramento. A comunicação assíncrona de SCA (Service Component Architecture), importações e exportações JMS e processamento interno assíncrono utilizam filas de mensagens no mecanismo do sistema de mensagens.

O ambiente de implementação conecta a origem da mensagem ao destino da mensagem através do barramento quando os módulos do aplicativo são implementados. Saber a origem da mensagem e o destino da mensagem ajuda você a determinar de que tipo de ambiente de implementação você precisa.

Aplicativos podem armazenar dados persistentes em um armazém de dados, que é um conjunto de tabelas em um banco de dados ou esquema, ou em um armazenamento de arquivos. O mecanismo do sistema de mensagens utiliza uma instância de uma origem de dados JDBC para interagir com esse banco de dados.

Configure o host de destino do sistema de mensagens ao definir seu ambiente de implementação utilizando o **Servidor** do console administrativo ou designe o servidor como o host de destino durante a instalação do software.

Armazém de Dados:

Cada mecanismo do sistema de mensagens pode utilizar um armazém de dados, que é um conjunto de tabelas em um banco de dados ou esquema que armazena dados persistentes.

Todas as tabelas no armazém de dados são mantidas no mesmo esquema do banco de dados. Você pode criar cada armazém de dados em um banco de dados separado. Como alternativa, pode criar vários armazéns de dados no mesmo banco de dados, com cada armazém de dados utilizando um esquema diferente.

Um mecanismo do sistema de mensagens utiliza uma instância de uma origem de dados JDBC para interagir com o banco de dados que contém o armazém de dados para esse mecanismo do sistema de mensagens.

Provedores JDBC:

Você pode usar provedores JDBC para que haja interação dos aplicativos com os bancos de dados relacionais.

Os aplicativos utilizam provedores JDBC para interagir com os bancos de dados relacionais. O provedor JDBC fornece a classe de implementação específica do driver JDBC para acesso a um tipo de banco de dados específico. Para criar um conjunto de conexões para esse banco de dados, você associa uma origem de dados ao provedor JDBC. Em conjunto, os objetos do provedor JDBC e da origem de dados são funcionalmente equivalentes ao connection factory de Java EE Connector Architecture (JCA), que fornece conectividade com um banco de dados não relacional.

Consulte os exemplos da configuração de ambiente independente Típica e da configuração de ambiente de implementação Típica no tópico anterior.

Para obter mais informações sobre provedores JDBC, consulte “Provedores JDBC” no centro de informações do WebSphere Application Server.

Barramento de Integração de Serviço para IBM Business Process Manager:

Um barramento de integração de serviços é um mecanismo de comunicação gerenciada que suporta a integração de serviços por meio de sistemas de mensagens síncronos e assíncronos. Um barramento consiste na interconexão de mecanismos de sistemas de mensagens que gerenciam recursos de barramento. É uma das tecnologias do WebSphere Application Server na qual o IBM Business Process Manager é baseado.

Um único barramento de integração de serviço e um único mecanismo do sistema de mensagens usam o mesmo esquema de banco de dados como o banco de dados do produto por padrão. Cada ambiente de implementação tem seu próprio barramento. Esse barramento único é chamado

`BPM.deployment_environment_name.Bus`.

Um destino de barramento é um endereço lógico ao qual os aplicativos podem conectar-se como um produtor, consumidor ou ambos. Um destino de fila é um destino de barramento utilizado para o sistema de mensagens ponto a ponto.

Cada barramento pode ter um ou mais membros de barramento, cada um sendo um servidor ou um cluster.

A *topologia de barramento* é a organização física dos servidores de aplicativos, mecanismos do sistema de mensagens e gerenciadores de fila do WebSphere MQ e o padrão de conexões de barramento e links entre elas, que constitui seu barramento de serviço corporativo.

Aplicativos de Serviço e Módulos de Serviço

Um módulo de serviço é um módulo de Service Component Architecture (SCA) que fornece serviços no ambiente de tempo de execução. Ao implementar um módulo de serviço no IBM Business Process Manager, construa um aplicativo de serviço associado empacotado como um arquivo enterprise archive (EAR).

Os módulos de serviço são as unidades básicas de implementação e podem conter componentes, bibliotecas e módulos de temporariedade usados pelo aplicativo de serviço associado. Os módulos de serviço possuem exportações e, opcionalmente, importações para definir os relacionamentos entre módulos e solicitantes e provedores de serviço. O WebSphere Process Server suporta módulos para serviços de negócios e módulos de mediação. Ambos os módulos e módulos de mediação são tipos de módulos SCA. Um módulo de mediação permite a comunicação entre aplicativos transformando a chamada de serviço em um formato entendido pelo destino, transmitindo a solicitação ao destino e retornando o resultado ao originador. Um módulo para um serviço de negócios implementa a lógica de um processo de negócios. Entretanto, um módulo também pode incluir a mesma lógica de mediação que pode ser empacotada em um módulo de mediação.

Implementando um Aplicativo de Serviço

O processo de implementação de um arquivo EAR que contém um aplicativo de serviços é o mesmo que o processo de implementação de qualquer arquivo EAR. É possível modificar os valores para os parâmetros de mediação no momento da implementação. Após ter implementado um arquivo EAR contendo um módulo SCA, você pode visualizar detalhes sobre o aplicativo de serviço e seu módulo associado. É possível ver como um módulo de serviço é conectado aos solicitantes de serviços (por meio de exportações) e provedores de serviços (por meio de importações).

Visualizando Detalhes do Módulo SCA

Os detalhes do módulo de serviço que você pode visualizar dependem do módulo SCA. Eles incluem os seguintes atributos.

- Nome do módulo SCA
- Descrição do módulo SCA
- Nome do aplicativo associado
- Informações sobre a versão do módulo SCA, se o módulo estiver com versão
- Importações do módulo SCA:
 - Interfaces de importação são definições abstratas que descrevem como um módulo SCA acessa um serviço.
 - Ligações de importação são definições concretas que especificam o mecanismo físico pelo qual um módulo SCA acessa um serviço. Por exemplo, usando SOAP/HTTP.
- Exportações de módulo SCA:
 - Interfaces de exportação são definições abstratas que descrevem como solicitantes de serviços acessam um módulo SCA.
 - Ligações de exportação são definições concretas que especificam o mecanismo físico pelo qual um solicitante de serviços acessa um módulo SCA e, indiretamente, um serviço.
- Propriedades do Módulo SCA

Importações e Ligações de Importações:

As importações definem as interações entre os módulos SCA e os fornecedores de serviços. Os módulos SCA usam importações para permitir que componentes acessem serviços externos (serviços que estão fora do módulo SCA) usando uma representação local. Ligações de importação definem a forma específica que um serviço externo é acessado.

Se os módulos SCA não precisarem acessar serviços externos, eles não precisarão ter importações. Os módulos de mediação normalmente uma ou mais importações que são utilizadas para passar mensagens ou solicitações a seus destinos pretendidos.

Interfaces e Ligações

Uma importação do módulo SCA precisa de pelo menos uma interface e uma importação do módulo SCA possui uma única ligação.

- Interfaces de importação são definições abstratas que definem um conjunto de operações usando Web Services Description Language (WSDL), uma linguagem XML para descrever serviços da web. Um módulo SCA pode ter muitas interfaces de importação.
- As ligações de importação são definições concretas que especificam o mecanismo físico que os módulos SCA usam para acessar um serviço externo.

Ligações de Importação Suportadas

O IBM Business Process Manager suporta as seguintes ligações de importação:

- As ligações SCA conectam os módulos SCA a outros módulos SCA. As ligações SCA também são referenciadas como ligações padrão.
- As ligações de serviço da web permitem que os componentes chamem serviços da web. Os protocolos suportados são SOAP1.1/HTTP, SOAP1.2/HTTP e SOAP1.1/JMS.

É possível usar uma ligação SOAP1.1/HTTP ou SOAP1.2/HTTP com base em Java API para XML Web Services (JAX-WS), que permite a interação com serviços usando ligações literais de documento ou de RPC e que usa manipuladores JAX-WS para customizar as chamadas. Uma ligação SOAP1.1/HTTP

separada é fornecida para permitir interação com serviços que usam uma ligação codificada por RPC ou em que há um requisito para usar manipuladores JAX-RPC para customizar chamadas.

- As ligações HTTP permitem que você acesse aplicativos usando o protocolo HTTP.
- As ligações de importação Enterprise JavaBeans (EJB) permitem que componentes SCA chamem serviços fornecidos pela lógica de negócios Java EE em execução em um servidor Java EE.
- As ligações de Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Essa comunicação é feita por meio do uso de adaptadores de recursos.
- As ligações do Java Message Service (JMS) 1.1 permitem interoperabilidade com o provedor de sistema de mensagem padrão do WebSphere Application Server. O JMS pode explorar vários tipos de transporte, incluindo TCP/IP e HTTP ou HTTPS. A classe de Mensagem JMS e seus cinco subtipos (Texto, Bytes, Objeto, Fluxo e Mapa) são suportados automaticamente.
- As ligações JMS genéricas permitem interoperabilidade com provedores JMS de terceiros que são integrados com o WebSphere Application Server usando o JMS Application Server Facility (ASF).
- As ligações JMS do WebSphere MQ permitem interoperabilidade com os provedores JMS baseados no WebSphere MQ. A classe de Mensagem JMS e seus cinco subtipos (Texto, Bytes, Objeto, Fluxo e Mapa) são suportados automaticamente. Se desejar usar o WebSphere MQ como um provedor JMS, use as ligações JMS do WebSphere MQ.
- As ligações do WebSphere MQ permitem interoperabilidade com o WebSphere MQ. É possível usar as ligações do WebSphere MQ apenas com gerenciadores de fila remotos por meio de uma conexão do cliente do WebSphere MQ; não é possível usá-las com gerenciadores de filas locais. Use as ligações do WebSphere MQ se desejar se comunicar com aplicativos WebSphere MQ nativos.

Chamada Dinâmica de Serviços

Os serviços podem ser chamados por meio de qualquer ligação de importação suportada. Um serviço está normalmente localizado em um terminal especificado na importação. Este terminal é chamado de terminal estático. É possível chamar um serviço diferente substituindo o terminal estático. A substituição dinâmica de terminais estáticos permite chamar um serviço em outro terminal, por meio de qualquer ligação de importação suportada. A chamada dinâmica de serviços também permite que você chame um serviço onde a ligação de importação suportada não tem um terminal estático.

Uma importação com uma ligação associada é usada para especificar o protocolo e sua configuração para chamada dinâmica. A importação usada para a chamada dinâmica pode ser ligada ao componente de chamada ou pode ser selecionada dinamicamente no tempo de execução.

Para chamadas de serviço da web e SCA, também é possível fazer uma chamada dinâmica sem uma importação, com o protocolo e a configuração deduzidos da URL de terminal. O tipo de destino da chamada é identificado a partir da URL do terminal. Se uma importação for usada, a URL deve ser compatível com o protocolo da ligação de importação.

- Uma URL SCA indica a chamada de um outro módulo SCA.
- Um URL de HTTP ou JMS por padrão indica a chamada de um serviço da web; para essas URLs, é possível fornecer um valor de tipo de ligação adicional que indica que a URL representa uma chamada por meio de uma ligação HTTP ou JMS.
- Para uma URL de HTTP de serviço da web, o padrão é usar SOAP 1.1 e pode ser especificado um valor de tipo de ligação que indica o uso de SOAP 1.2.

Exportações e Ligações de Exportações:

As exportações definem as interações entre os módulos SCA e os solicitantes de serviços. Os módulos SCA usam exportações para oferecer serviços para outros. As ligações de exportação definem a maneira específica pela qual um módulo SCA é acessado pelos solicitantes de serviço.

Interfaces e Ligações

Uma exportação de módulo SCA precisa de, pelo menos, uma interface.

- Interfaces de exportação são definições abstratas que definem um conjunto de operações usando Web Services Description Language (WSDL), uma linguagem XML para descrever serviços da web. Um módulo SCA pode ter muitas interfaces de exportação.
- As ligações de exportação são definições concretas que especificam o mecanismo físico que os solicitantes de serviço usam para acessar um serviço. Geralmente, uma exportação de módulo SCA possui uma ligação especificada. Uma exportação sem ligação especificada é interpretada pelo ambiente de tempo de execução como uma exportação com uma ligação SCA.

Ligações de Exportação Suportadas

O IBM Business Process Manager suporta as seguintes ligações de exportação:

- As ligações SCA conectam os módulos SCA a outros módulos SCA. As ligações SCA também são referenciadas como ligações padrão.
- As ligações de serviço da web permitem que as exportações sejam chamadas como serviços da web. Os protocolos suportados são SOAP1.1/HTTP, SOAP1.2/HTTP e SOAP1.1/JMS.
É possível usar uma ligação SOAP1.1/HTTP ou SOAP1.2/HTTP com base na Java API for XML Web Services (JAX-WS), que permite interação com os serviços usando ligações do documento ou de literal do RPC e usa os manipuladores JAX-WS para customizar as chamadas. Uma ligação SOAP1.1/HTTP separada é fornecida para permitir interação com serviços que usam uma ligação codificada por RPC ou em que há um requisito para usar manipuladores JAX-RPC para customizar chamadas.
- As ligações HTTP permitem que as exportações sejam acessadas usando o protocolo HTTP.
- As ligações de exportação do Enterprise JavaBeans (EJB) permitem que os componentes do SCA sejam expostos como EJBs para que a lógica de negócios do Java EE possa chamar os componentes do SCA ou, do contrário, indisponibilizá-los.
- As ligações de Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Essa comunicação é feita por meio do uso de adaptadores de recursos.
- As ligações do Java Message Service (JMS) 1.1 permitem interoperabilidade com o provedor de sistema de mensagem padrão do WebSphere Application Server. O JMS pode explorar vários tipos de transporte, incluindo TCP/IP e HTTP ou HTTPS. A classe de Mensagem JMS e seus cinco subtipos (Texto, Bytes, Objeto, Fluxo e Mapa) são suportados automaticamente.
- As ligações JMS genéricas permitem interoperabilidade com provedores JMS de terceiros que são integrados com o WebSphere Application Server usando o JMS Application Server Facility (ASF).
- As ligações JMS do WebSphere MQ permitem interoperabilidade com os provedores JMS baseados no WebSphere MQ. A classe de Mensagem JMS e seus cinco subtipos (Texto, Bytes, Objeto, Fluxo e Mapa) são suportados automaticamente. Se desejar usar o WebSphere MQ como um provedor JMS, use as ligações JMS do WebSphere MQ.
- As ligações do WebSphere MQ permitem interoperabilidade com o WebSphere MQ. Use uma conexão remota (ou do cliente) para se conectar com um gerenciador de filas do MQ em uma máquina remota. Uma conexão (ou ligações) local é uma conexão direta com o WebSphere MQ. Isso pode ser usado apenas para uma conexão com um gerenciador de filas do MQ na mesma máquina. O WebSphere MQ permitirá os dois tipos de conexão, mas as ligações do MQ suportarão apenas a conexão "remota" (ou do "cliente").

Módulos de Mediação:

Os módulos de mediação são módulos Service Component Architecture (SCA) que podem alterar o formato, o conteúdo ou o destino de solicitações de serviço.

Os módulos de mediação operam em mensagens que estão em andamento entre solicitantes de serviços e provedores de serviços. É possível rotear mensagens para diferentes provedores de serviços e aditar o

conteúdo da mensagem ou formulário. Os módulos de mediação podem fornecer funções como criação de log de mensagens e processamento de erro que são customizadas de acordo com seus requisitos.

É possível alterar determinados aspectos de módulos de mediação, a partir do console administrativo, sem precisar reimplementar o módulo.

Componentes dos Módulos de Mediação

Módulos de mediação contêm os seguintes itens:

- Importações, que definem as interações entre os módulos SCA e os provedores de serviços. Elas permitem que os módulos SCA chamem serviços externos como se fossem locais. É possível visualizar as importações do módulo de mediação e modificar a ligação.
- Exportações, que definem as interações entre os módulos SCA e os solicitantes de serviços. Elas permitem que um módulo SCA ofereça um serviço e defina as interfaces externas (pontos de acesso) de um módulo SCA. É possível visualizar as exportações do módulo de mediação.
- Componentes SCA, que são blocos de construção para módulos SCA, tais como módulos de mediação. É possível criar e customizar módulos SCA e componentes graficamente, usando Integration Designer. Depois de implementar um módulo de mediação, é possível customizar determinados aspectos dele a partir do console administrativo, sem precisar reimplementar o módulo.

Geralmente, os módulos de mediação contêm um tipo específico de componente SCA chamado de *componente de fluxo de mediação*. Os componentes do fluxo de mediação definem fluxos de mediação.

Um componente do fluxo de mediação pode conter nenhuma, uma ou várias primitivas de mediação. O IBM Business Process Manager suporta um conjunto fornecido de primitivas de mediação que fornecem funcionalidade para roteamento e transformação de mensagem. Para flexibilidade adicional da primitiva de mediação, use a Primitiva de mediação customizada para chamar a lógica customizada.

O propósito de um módulo de mediação que não contém um componente do fluxo de mediação é transformar solicitações de serviço de um protocolo para outro. Por exemplo, um pedido de serviço pode ser feito utilizando SOAP/JMS mas pode precisar ser transformado para SOAP/HTTP antes do envio.

Nota: É possível visualizar e fazer determinadas alterações nos módulos de mediação a partir do IBM Business Process Manager. No entanto, não é possível visualizar ou alterar os componentes SCA dentro de um módulo a partir do IBM Business Process Manager. Use o Integration Designer para customizar componentes SCA.

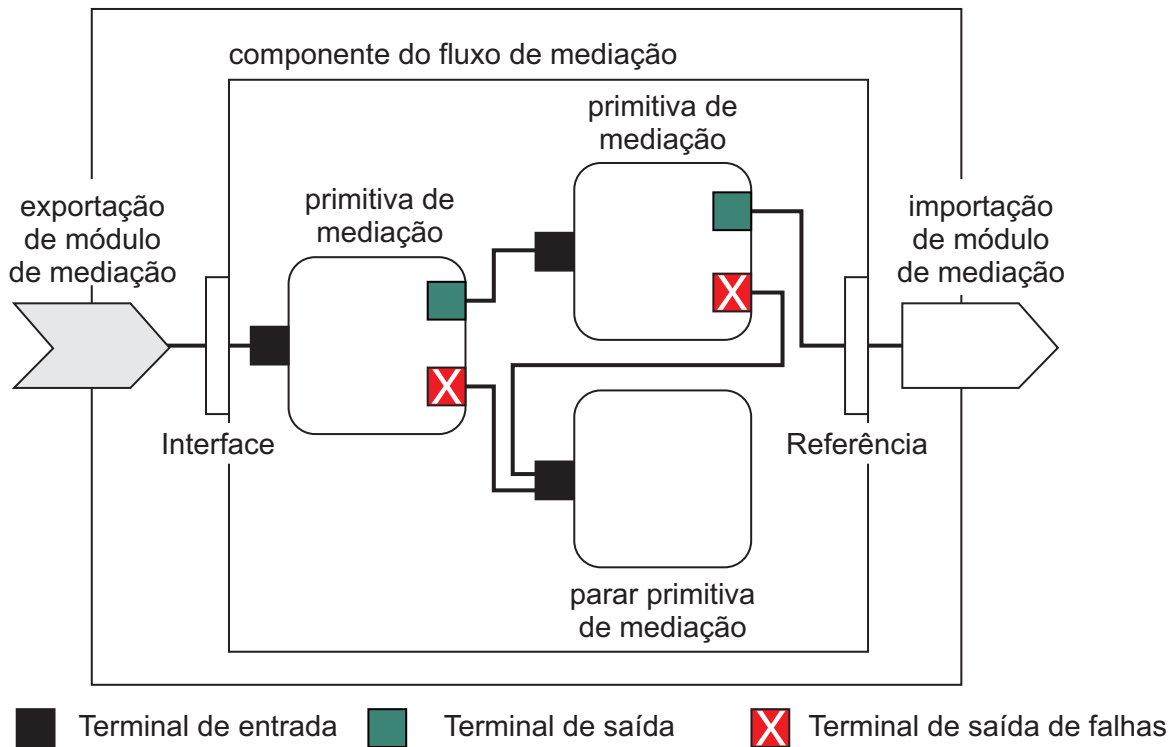


Figura 43. Exemplo simplificado de um módulo de mediação. O módulo de mediação contém um componente do fluxo de mediação, que contém primitivas de mediação.

- Propriedades

As primitivas de mediação possuem propriedades, algumas das quais podem ser exibidas no console administrativo como propriedades adicionais de um módulo SCA.

Para propriedades da primitiva de mediação ficarem visíveis a partir do console administrativo do IBM Business Process Manager, o desenvolvedor de integração deve promover as propriedades.

Determinadas propriedades aperfeiçoam-se sendo administrativamente configuradas e o Integration Designer as descreve como propriedades promovíveis, pois elas podem ser promovidas a partir do ciclo de integração para o ciclo administrativo. Outras propriedades não são adequadas para configuração administrativa, porque a modificação delas pode afetar o fluxo de mediação de tal maneira que o módulo de mediação precisa ser reimplementado. O Integration Designer lista as propriedades que você pode escolher para promover nas propriedade promovidas de uma primitiva de mediação.

É possível usar o console administrativo do IBM Business Process Manager para alterar o valor das propriedades promovidas sem precisar reimplementar um módulo de mediação ou reiniciar o servidor ou módulo.

Geralmente, os fluxos de mediação usam as alterações de propriedade imediatamente. No entanto, se as mudanças de propriedade ocorrerem em uma célula do gerenciador de implementação, elas terão efeito em cada nó conforme esse nó for sincronizado. Além disso, os fluxos de mediação que estão em andamento continuam utilizando valores anteriores.

Nota: No console administrativo, você só pode alterar os valores da propriedade, não grupos de propriedades, nomes ou tipos. Se quiser alterar grupos de propriedades, nomes ou tipos, você deve usar o Integration Designer.

- Um módulo de mediação ou biblioteca dependente também pode definir os subfluxos. Um subfluxo contém um conjunto de primitivas de mediação ligadas como uma parte reutilizável da lógica de integração. Uma primitiva pode ser incluída em um fluxo de mediação para chamar um subfluxo.

Implementando Módulos de Mediação

Os módulos de mediação são criados usando o Integration Designer e são geralmente implementados no IBM Business Process Manager dentro de um arquivo de archive corporativo (EAR).

Você pode alterar o valor das propriedades promovidas no momento da implementação.

É possível exportar um módulo de mediação a partir do Integration Designer e fazer com que o Integration Designer empacote o módulo de mediação dentro de um arquivo Java archive (JAR) e o arquivo JAR dentro de um arquivo EAR. Você pode, então, implementar o arquivo EAR, instalando um novo aplicativo a partir do console administrativo.

Os módulos de mediação podem ser considerado como uma entidade. No entanto, os módulos SCA são definidos por vários arquivos XML armazenados em um arquivo JAR.

Exemplo de arquivo EAR, contendo um módulo de mediação

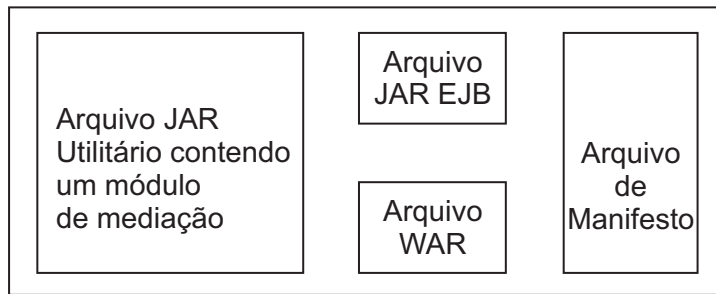


Figura 44. Exemplo simplificado de um arquivo EAR contendo um módulo de mediação. O arquivo EAR contém JARs. O arquivo JAR utilitário contém um módulo de mediação.

Primitivas de Mediação:

Os componentes do fluxo de mediação operam em fluxos de mensagens entre componentes de serviço. Os recursos de um componente de mediação são implementados pelas *primitivas de mediação*, que implementam tipos de implementação de serviço padrão.

Um componente do fluxo de mediação possui um ou mais fluxos. Por exemplo, um para pedido e um para resposta.

O IBM Business Process Manager suporta um conjunto fornecido de primitivas de mediação, o qual implementa recursos de mediação padrão para módulos de mediação ou módulos implementados no IBM Business Process Manager. Se precisar de recursos de mediação especiais, será possível desenvolver suas próprias primitivas de mediação customizadas.

Uma primitiva de mediação define uma operação de “entrada” que processa ou manipula mensagens que são representadas por SMOs (Service Message Objects). Uma primitiva de mediação também pode definir operações de “saída” que enviam mensagens para outro componente ou módulo.

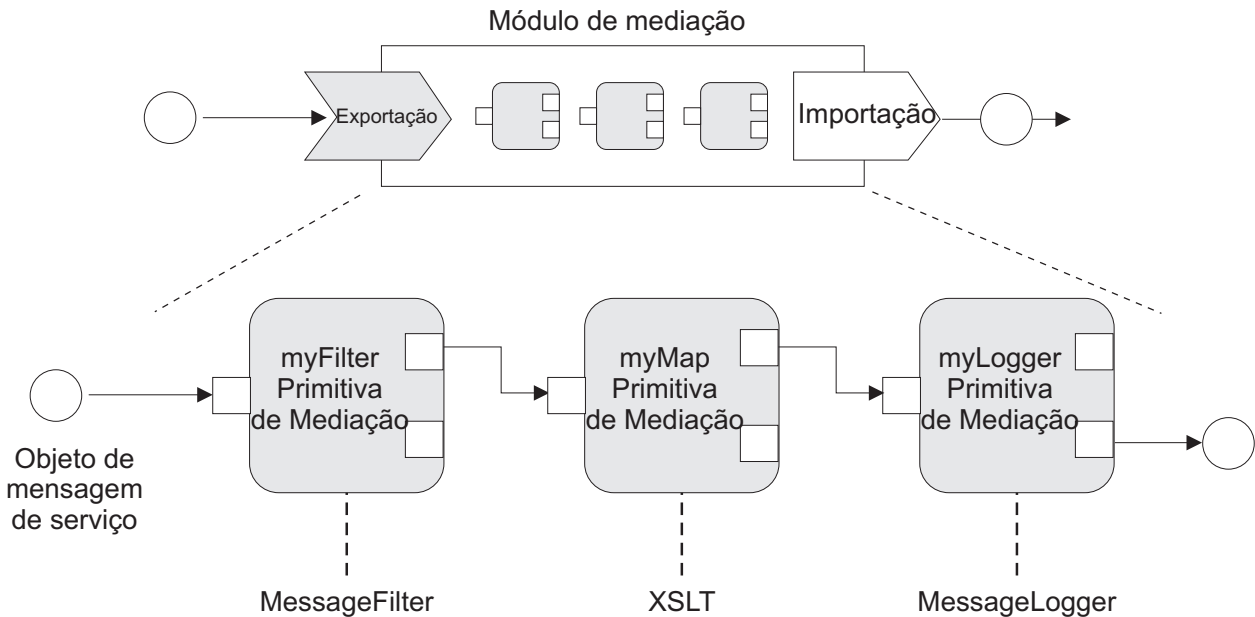


Figura 45. Módulo de Mediação Contendo Três Primitivas de Mediação

Você pode utilizar o Integration Designer para configurar primitivas de mediação e configurar suas propriedades. Algumas destas propriedades podem ficar visíveis para o administrador de tempo de execução sendo promovidas. Qualquer propriedade de primitiva de mediação que pode ser promovida também pode ser uma propriedade dinâmica. Uma propriedade dinâmica pode ser substituída, no tempo de execução, utilizando um arquivo de políticas.

O Integration Designer também permite modelar e montar graficamente componentes do fluxo de mediação a partir de primitivas de mediação e montar módulos de mediação ou módulos a partir dos componentes do fluxo de mediação. O console administrativo refere-se a módulos de mediação e módulos como módulos SCA.

Integration Designer também permite a definição de subfluxos nos módulos ou em suas bibliotecas dependentes. Um subfluxo pode conter qualquer primitiva de mediação, exceto para a primitiva de mediação Resolução de Política. Um subfluxo é chamado a partir de um fluxo de pedido ou resposta ou de outro subfluxo utilizando a primitiva de mediação Subfluxo. As propriedades promovidas a partir das primitivas de mediação em um subfluxo são expostas como propriedades nas primitivas de mediação Subfluxo. Elas podem, então, ser promovidas novamente até que alcancem o nível do módulo; nesse ponto elas poderão ser modificadas pelo administrador de tempo de execução.

Primitivas de Mediação Suportadas

O conjunto de primitivas de mediação a seguir é suportado pelo IBM Business Process Manager:

Mapa de Objetos de Negócios

Transforma mensagens.

- Define transformações de mensagens utilizando um mapa de objetos de negócios, que pode ser reutilizado.
- Permite definir transformações de mensagens graficamente, utilizando o editor de mapas de objetos de negócios.
- Pode alterar o conteúdo de uma mensagem.
- Pode transformar um tipo de mensagem de entrada em um tipo de mensagem de saída diferente.

Mediação Customizada

Permite implementar sua própria lógica de mediação no código Java. A primitiva de Mediação Customizada combina a flexibilidade de uma primitiva de mediação definida pelo usuário, com a simplicidade de uma primitiva de mediação predefinida. É possível criar transformações complexas e padrões de roteamento:

- Criando código Java.
- Criando suas próprias propriedades.
- Incluindo novos terminais.

É possível chamar um serviço a partir de uma primitiva de Mediação Customizada, mas a primitiva de mediação Chamada de Serviço foi projetada para chamar serviços e fornece funcionalidade adicional, como uma nova tentativa.

Manipulador de Dados

Permite transformar uma parte de uma mensagem. É utilizado para converter um elemento de uma mensagem de um formato físico em uma estrutura lógica ou de uma estrutura lógica em um formato físico. O principal uso da primitiva é converter um formato físico, como uma cadeia de Texto em um objeto de Mensagem de Texto JMS, em uma estrutura lógica do Objeto de Negócios e fazer a conversão de volta. Esta mediação é comumente utilizada para:

- Transforme uma seção da mensagem de entrada de uma estrutura definida em outra - um exemplo seria quando o SMO inclui um valor de sequência que é delimitado por vírgula e deseja analisar para um Objeto de Negócios específico.
- Alterar o tipo de mensagem – um exemplo disto seria quando uma exportação JMS tiver sido configurada para utilizar uma ligação de dados básica categorizada por JMS e, no módulo de mediação, o desenvolvedor de integração decide se o conteúdo deve ser expandido para uma estrutura de BO específica.

Consulta ao Banco de Dados

Modifica mensagens, utilizando informações de um banco de dados fornecido pelo usuário.

- Você deve configurar um banco de dados, uma origem de dados e quaisquer configurações de autenticação de servidor para a primitiva de mediação de Consulta do Banco de Dados usar. Utilize o console administrativo para ajudá-lo a fazer isso.
- A primitiva de mediação de Consulta do Banco de Dados pode ler a partir apenas de uma tabela.
- A coluna-chave especificada deve conter um valor exclusivo.
- Os dados nas colunas de valor devem ter um tipo de esquema XML simples ou um tipo de esquema XML que estende um tipo de esquema XML simples.

Consulta do Terminal

Permite o roteamento dinâmico de pedidos, procurando terminais de serviço em um repositório.

- As informações do terminal em serviço são recuperadas de um WebSphere Service Registry and Repository (WSRR). O registro WSRR pode ser local ou remoto.
- Você faz alterações no registro a partir do console administrativo do WSRR.
- O IBM Business Process Manager precisa saber qual registro utilizar, portanto, é necessário criar definições de acesso de WSRR utilizando o console administrativo do IBM Business Process Manager.

Emissor de Eventos

Melhora o monitoramento permitindo o envio de eventos de dentro de um componente do fluxo de mediação.

- É possível suspender a ação de mediação ao desmarcar a caixa de opção.
- É possível visualizar os eventos do Emissor de Eventos usando o navegador do Common Base Events no IBM Business Process Manager.
- Você deve enviar apenas eventos em um ponto significativo em um fluxo de mediação, por motivos de desempenho.

- É possível definir as partes da mensagem contidas no evento.
- Os eventos são enviados em forma de Common Base Events e são enviados para um servidor Common Event Infrastructure.
- Para utilizar totalmente as informações do Emissor de Eventos, os consumidores de eventos precisam entender a estrutura dos Common Base Events. Os Common Base Events possuem um esquema geral, mas isto não modela os dados específicos do aplicativo, que estão contidos nos elementos de dados estendidos. Para modelar os elementos de dados estendidos, as ferramentas do Integration Designer geram um arquivo de definição do catálogo de eventos de Common Event Infrastructure para cada uma das primitivas de mediação do Emissor de Evento configurado. Os arquivos de definição de catálogo de eventos são artefatos de exportação fornecidos para ajudá-lo; eles não são usados pelo Integration Designer ou pelo tempo de execução do IBM Business Process Manager. Você deve consultar os arquivos de definição do catálogo de eventos ao criar aplicativos para consumir eventos do Emissor de Eventos.
- Você pode especificar outro monitoramento no IBM Business Process Manager. Por exemplo, você pode monitorar eventos a serem emitidos a partir de importações e exportações.

Falha Pára um caminho específico no fluxo e gera uma exceção.

Fan In Ajuda a agregar (combinar) mensagens.

- Pode ser utilizado apenas em combinação com a primitiva de mediação Fan Out.
- Juntas, as primitivas de mediação Fan Out e Fan In permitem a agregação de dados em uma mensagem de saída.
- A primitiva de mediação Fan In recebe mensagens até que um ponto de decisão seja alcançado, em seguida, ocorre uma saída de mensagem.
- O contexto compartilhado deve ser utilizado para conter dados de agregação.

Fan Out

Ajuda a dividir e agregar (combinar) mensagens.

- Juntas, as primitivas de mediação Fan Out e Fan In permitem a agregação de dados em uma mensagem de saída.
- No modo de iteração, a primitiva de mediação Fan Out permite iterar por meio de uma única mensagem de entrada que contém o elemento de repetição. Para cada ocorrência do elemento de repetição, é enviada uma mensagem.
- O contexto compartilhado deve ser utilizado para conter dados de agregação.

Configurador do Cabeçalho HTTP

Fornece um mecanismo para gerenciar cabeçalhos em mensagens HTTP.

- Pode criar, configurar, copiar ou excluir cabeçalhos da mensagem HTTP.
- Pode configurar várias ações para alterar vários cabeçalhos HTTP.

Mapeamento

Transforma mensagens.

- Permite executar transformações Linguagem de Folha de Estilo Extensível (XSL) ou de Mapa de Objetos de Negócios.
- Você transforma mensagens usando uma transformação XSLT 1.0 ou XSLT 2.0 ou Mapa de Objeto de Negócios. As transformações XSL operam em uma serialização XML da mensagem, enquanto que a transformação Mapa de Objeto de Negócios opera no Service Data Objects (SDO).

Configurador do Elemento de Mensagem

Fornece um mecanismo simples de configurar o conteúdo de mensagens.

- Pode alterar, incluir ou excluir elementos de mensagem.
- Não altera o tipo da mensagem.

- Os dados nas colunas de valor devem ter um tipo de esquema XML simples ou um tipo de esquema XML que estende um tipo de esquema XML simples.

Filtro de Mensagem

Roteia mensagens por diferentes caminhos, com base no conteúdo da mensagem.

- É possível suspender a ação de mediação ao desmarcar a caixa de opção.

Registrador de Mensagem

Registra mensagens em um banco de dados relacional ou por meio de seu próprio criador de log. As mensagens são armazenadas como XML, portanto, os dados podem ser pós-processados por aplicativos identificados por XML.

- É possível suspender a ação de mediação ao desmarcar a caixa de opção.
- O esquema do banco de dados relacional (estrutura de tabela) é definido pela IBM.
- Por padrão, a primitiva de mediação Message Logger utiliza o banco de dados Comum. O ambiente de tempo de execução mapeia a origem de dados em `jdbc/mediation/messageLog` para o banco de dados Comum.
- Você pode configurar classes de implementação do Manipulador para customizar o comportamento do criador de log customizado. Opcionalmente, é possível fornecer classes de implementação do Formatador, classes de implementação do Filtro ou ambas para customizar o comportamento do criador de log customizado.

Configurador do Cabeçalho do MQ

Fornece um mecanismo para gerenciar cabeçalhos em mensagens do MQ.

- Pode criar, configurar, copiar ou excluir cabeçalhos da mensagem do MQ.
- Pode configurar várias ações para alterar vários cabeçalhos do MQ.

Resolução de Política

Permite a configuração dinâmica de pedidos, procurando terminais em serviço e arquivos de políticas associados, em um repositório.

- É possível utilizar um arquivo de políticas para substituir dinamicamente as propriedades promovidas de outras primitivas de mediação.
- As informações do terminal em serviço e as informações de política são recuperadas de um WebSphere Service Registry and Repository (WSRR). O registro WSRR pode ser local ou remoto.
- Você faz alterações no registro a partir do console administrativo do WSRR.
- O IBM Business Process Manager precisa saber qual registro utilizar, portanto, é necessário criar definições de acesso de WSRR utilizando o console administrativo do IBM Business Process Manager.

Chamada de Serviço

Chama um serviço a partir de um fluxo de mediação em vez de aguardar até o final do fluxo de mediação e utiliza o mecanismo callout.

- Se o serviço retornar uma falha, você poderá tentar novamente o mesmo serviço ou chamar outro serviço.
- A primitiva de mediação Chamada de Serviço é uma poderosa primitiva de mediação que pode ser utilizada para chamadas de serviço simples, ou em combinação com outras primitivas de mediação para mediações complexas.

Tipo de Mensagem Configurado

Durante o desenvolvimento de integração, permite tratar campos de mensagens de texto simples como se fossem de texto restrito. Um campo é de texto simples se puder conter mais de um tipo de dados. Um campo é de texto restrito se seu tipo de estrutura interna forem conhecidos.

- No tempo de execução, a primitiva de mediação Tipo de Mensagens permite verificar se o conteúdo de uma mensagem corresponde aos tipos de dados esperados.

Configurador do Cabeçalho SOAP

Fornece um mecanismo para gerenciar cabeçalhos em mensagens SOAP.

- Pode criar, configurar, copiar ou excluir cabeçalhos da mensagem SOAP.
- Pode configurar várias ações para alterar vários cabeçalhos SOAP.

Parar Pára um caminho específico no fluxo, sem gerar uma exceção.

Filtro de Tipo

Permite direcionar mensagens para um caminho diferente de um fluxo, com base em seu tipo.

WebSphere eXtreme Scale Recuperar

É possível recuperar informações de um ambiente de cache do servidor do eXtreme Scale.

- É possível consultar valores no cache e armazená-los como elementos na mensagem usando uma chave.
- Combinando as primitivas de mediação eXtreme Scale Armazenar e Recuperar, é possível armazenar em cache a resposta de um sistema backend. Solicitações futuras não exigirão acesso a esse sistema backend.
- Você deve criar definições do eXtreme Scale usando o console administrativo WebSphere ESB, portanto será possível especificar qual servidor eXtreme Scale usar.

WebSphere eXtreme Scale Armazenar

É possível armazenar informações em um ambiente de cache do servidor eXtreme Scale.

- É possível armazenar informações em um cache do eXtreme Scale usando uma chave e um objeto.
- Combinando as primitivas de mediação eXtreme Scale Armazenar e Recuperar, é possível usar a primitiva de mediação Armazenar para armazenar dados no cache e usar a primitiva de mediação Recuperar para recuperar dados armazenados anteriormente na cache.
- Você deve criar definições do eXtreme Scale usando o console administrativo WebSphere ESB, portanto será possível especificar qual servidor eXtreme Scale usar.

Roteamento Dinâmico:

É possível rotear mensagens de várias maneiras usando terminais definidos no momento da integração ou terminais determinados, dinamicamente, no tempo de execução.

Roteamento dinâmico abrange dois casos de roteamento de mensagem:

- Roteamento de mensagem em que o fluxo é dinâmico, mas todos os terminais possíveis são predefinidos em um módulo de Service Component Architecture (SCA).
- Roteamento de mensagem em que o fluxo é dinâmico e a seleção de terminal também é dinâmica. Os terminais de serviço são selecionados de uma origem externa no tempo de execução.

Seleção de Terminal Dinâmico

O tempo de execução tem a capacidade de rotear as mensagens de solicitação e resposta para um endereço de terminal identificado por um elemento de cabeçalho da mensagem. Este elemento de cabeçalho da mensagem pode ser atualizado pelas primitivas de mediação, em um fluxo de mediação. O endereço de terminal poderá ser atualizado com as informações de um registro, um banco de dados ou com informações da própria mensagem. O roteamento de mensagens de resposta se aplica apenas quando a resposta está sendo enviada por uma exportação JAX-WS de serviço da web.

Para que o tempo de execução implemente o roteamento dinâmico em uma solicitação ou resposta, o módulo SCA deve ter a propriedade Usar terminal dinâmico se configurado no cabeçalho da mensagem configurada. Os desenvolvedores de integração podem configurar a propriedade Usar terminal dinâmico se configurado no cabeçalho da mensagem ou poderão promovê-la (torná-la visível no tempo de execução), para que o administrador de tempo de execução possa configurá-la. É possível visualizar as propriedades do módulo na janela Propriedades do Módulo. Para ver a janela, clique em **Aplicativos** >

Módulos SCA > Propriedades do Módulo. O desenvolvedor de integração fornece nomes de alias de propriedades promovidas e esses são os nomes exibidos no console administrativo.

Registro

É possível usar o IBM WebSphere Service Registry and Repository (WSRR) para armazenar informações do terminal em serviço e, em seguida, criar módulos SCA para recuperar terminais a partir do registro do WSRR.

Quando desenvolver módulos SCA, será possível usar a primitiva de mediação de Consulta de Terminal para permitir que um fluxo de mediação consulte um registro do WSRR para um terminal em serviço ou um conjunto de terminais em serviço. Se um módulo SCA recuperar um conjunto de terminais, ele deverá usar uma outra primitiva de mediação para selecionar o preferencial.

Controle de Política de Mediação de Solicitações de Serviço:

É possível usar políticas de mediação para controlar fluxos de mediação entre solicitantes de serviços e provedores de serviços.

É possível controlar os fluxos de mediação usando políticas de mediação armazenadas no IBM WebSphere Service Registry and Repository (WSRR). A implementação de gerenciamento de política de serviço no WSRR é baseada no Web Services Policy Framework (WS-Policy).

Para controlar as solicitações de serviço usando políticas de mediação, você deve ter módulos adequados de Service Component Architecture (SCA) e documentos de política de mediação em seu registro do WSRR.

Como Conectar uma Política de Mediação a uma Solicitação de Serviço

Ao desenvolver um módulo SCA que precisar fazer uso de uma política de mediação, você deve incluir uma primitiva de mediação de Resolução de Política no fluxo de mediação. No tempo de execução, a primitiva de mediação de Resolução de Política obtém informações da política de mediação a partir do registro. Portanto, um módulo SCA deve conter um componente do fluxo de mediação para suportar o controle de solicitações de serviço da política de mediação.

No registro, é possível anexar uma ou mais políticas de mediação para um módulo SCA ou para um serviço de destino usado pelo módulo SCA. As políticas de mediação anexadas podem ser usadas (estão no escopo) para todas as mensagens de serviço processadas por esse módulo SCA. As políticas de mediação podem ter anexos de política que definem as condições. As condições de política de mediação permitem que políticas de mediação diferentes sejam aplicadas em contextos diferentes. Além disso, as políticas de mediação podem ter classificações, que podem ser usadas para especificar um estado de controle.

WebSphere Service Registry and Repository:

O produto WebSphere Service Registry and Repository (WSRR) permite armazenar, acessar e gerenciar informações sobre terminais de serviço e políticas de mediação. É possível usar o WSRR para tornar seus aplicativos de serviço mais dinâmicos e mais adaptáveis para alterar as condições de negócios.

Introdução

Os fluxos de mediação podem usar WSRR como um mecanismo de consulta dinâmico, fornecendo informações sobre terminais de serviço ou políticas de mediação.

Para configurar o acesso ao WSRR, você cria documentos de definição do WSRR utilizando o console administrativo. Uma alternativa é utilizar os comandos de administração do WSRR do cliente de script

wsadmin. As definições de WSRR e suas propriedades da conexão são o mecanismo usado para se conectar a uma instância de registro e recuperar um terminal em serviço ou uma política de mediação.

Terminais em Serviço

É possível utilizar o WSRR para armazenar informações sobre serviços que já utiliza, que planeja utilizar ou sobre os quais gostaria de estar ciente. Estes serviços podem estar em seus ou em outros sistemas. Por exemplo, um aplicativo poderia usar o WSRR para localizar o serviço mais apropriado para satisfazer suas necessidades funcionais e de desempenho.

Ao desenvolver um módulo SCA que precisa acessar terminais em serviço a partir do WSRR, você deve incluir uma primitiva de mediação de Consulta de Terminal no fluxo de mediação. No tempo de execução, a primitiva de mediação Consulta de Terminal obtém os terminais em serviço do registro.

Políticas de Mediação

Também é possível usar o WSRR para armazenar as informações sobre política de mediação. As políticas de mediação podem ajudar a controlar as solicitações de serviço, substituindo dinamicamente as propriedades do módulo. Se o WSRR contiver políticas de mediação que são anexadas a um objeto representando seu módulo SCA ou seu serviço de destino, as políticas de mediação poderão substituir as propriedades do módulo. Se desejar que políticas de mediação diferentes sejam aplicadas em diferentes contextos, é possível criar condições de política de mediação.

Nota: As políticas de mediação estão preocupadas com o controle dos fluxos de mediação e não com a segurança.

Ao desenvolver um módulo SCA que precisar fazer uso de uma política de mediação, você deve incluir uma primitiva de mediação de Resolução de Política no fluxo de mediação. No tempo de execução, a primitiva de mediação de Resolução de Política obtém informações da política de mediação a partir do registro.

WebSphere eXtreme Scale:

Ao usar o produto WebSphere eXtreme Scale (eXtreme Scale), é possível fornecer um sistema de armazenamento em cache que pode ser integrado a um aplicativo IBM Business Process Manager. Usando o eXtreme Scale com IBM Business Process Manager é possível melhorar os tempos de resposta de serviço e a confiabilidade, e fornecer uma funcionalidade de integração adicional.

eXtreme Scale atua como uma grade de dados elásticos, escaláveis e de memória interna. A grade de dados armazena em cache, particiona, replica e gerencia dinamicamente os dados do aplicativo e a lógica de negócios em diversos servidores. Com o eXtreme Scale, também é possível obter as qualidades do serviço como integridade transacional, alta disponibilidade e tempos de resposta previsíveis.

É possível usar fluxos de mediação para acessar a função de armazenamento em cache do eXtreme Scale incluindo as primitivas de mediação do WebSphere eXtreme Scale em seu fluxo. Ao desenvolver um módulo Service Component Architecture (SCA) que precisa armazenar informações em um cache do eXtreme Scale, você deve incluir a primitiva de mediação Armazenar do WebSphere eXtreme Scale no fluxo de mediação. Se deseja recuperar informações de um cache do eXtreme Scale, você deve incluir a primitiva de mediação Recuperar do WebSphere eXtreme Scale. Ao combinar as duas primitivas de mediação em um fluxo de mediação, é possível armazenar em cache a resposta de um sistema backend, para que as solicitações futuras possam recuperar a resposta do cache.

Para configurar o acesso ao eXtreme Scale, você deve criar uma definição do WebSphere eXtreme Scale usando o console administrativo. Como alternativa, é possível usar os comandos de administração do

WebSphere eXtreme Scale a partir do cliente de script wsadmin. Uma definição do eXtreme Scale é o mecanismo usado pelas primitivas de mediação Recuperar e Armazenar do WebSphere eXtreme Scale para se conectar a um servidor eXtreme Scale.

Clientes do Serviço de Mensagens

Os clientes do serviço de mensagens estão disponíveis para C/C++ e .NET para permitir que aplicativos não Java se conectem ao barramento de serviço corporativo.

O Message Service Clients for C/C++ and .NET fornece uma API chamada XMS que possui o mesmo conjunto de interfaces que a API do Sistema de Mensagens Java (JMS). O Cliente do Serviço de Mensagens para C/C++ contém duas implementações do XMS, uma para uso pelos aplicativos C e outra para uso pelos aplicativos C++. O Message Service Client for .NET contém uma implementação totalmente gerenciada do XMS, que pode ser usada por qualquer linguagem compatível com .NET.

É possível obter os Clientes do Serviço de Mensagens para .NET a partir do http://www-01.ibm.com/support/docview.wss?rs=0&q1=IA9H&uid=swg24011756&loc=en_US&cs=utf-8&cc=us&lang=en

É possível obter Clientes do Serviço de Mensagens para C/C++ a partir do http://www-01.ibm.com/support/docview.wss?rs=0&q1=ia94&uid=swg24007092&loc=en_US&cs=utf-8&cc=us&lang=en.

Também é possível instalar e usar o suporte a clientes Java EE do WebSphere Application Server Network Deployment, incluindo cliente de serviços da web, Clientes EJB e Cliente JMS.

Capítulo 2. Aprenda Mais sobre os Conceitos Chave

Use esta seção como um ponto de início para investigar as tecnologias usadas no IBM Business Process Manager e por ele.

Cenários de Autoria

Use cenários para entender e trabalhar com componentes e produtos da família de gerenciamento de processos de negócios.

Criação de Versões

O ciclo de vida de um aplicativo de processo começa com a criação do aplicativo de processo e continua por um ciclo de atualização, implementação, coimplementação, remoção de implementação e arquivamento do aplicativo de processo. *Versão* é um mecanismo usado para gerenciar o ciclo de vida do aplicativo de processo identificando exclusivamente as versões individuais do aplicativo de processo.

A maneira como essa versão funciona no IBM Business Process Manager depende do que você está implementando – um aplicativo de processo, implementado a partir do repositório no IBM Process Center ou um aplicativo corporativo implementado diretamente a partir do IBM Integration Designer.

Os aplicativos de processo e kits de ferramentas que você implementa em um ambiente de tempo de execução a partir do Process Center são, por padrão, com versão. Para aplicativos corporativos, você pode optar por criar a versão de módulos e bibliotecas no IBM Integration Designer.

Além disso, é possível criar versões de uma tarefa manual ou máquina de estado, para que diversas versões da tarefa ou máquina de estado possam coexistir no ambiente de tempo de execução.

Versão dos Aplicativos de Processo

A versão fornece a capacidade para que o ambiente de tempo de execução identifique capturas instantâneas no ciclo de vida de um aplicativo de processo e possa executar diversas capturas instantâneas simultaneamente em um servidor de processos.

Para entender como os aplicativos de processo são transformados em versão, é importante lembrar-se que um aplicativo de processo é um contêiner que contém vários artefatos usados no aplicativo de processo ou por ele (por exemplo, modelos de processo ou BPDs, referências ao kit de ferramentas, serviços ou rastreios). Qualquer versão é feita nesse nível de contêiner, não no nível dos artefatos individuais. Para aplicativos de processo, isso significa que a versão acontece quando uma captura instantânea é obtida.

É possível comparar capturas instantâneas para determinar diferenças entre as versões. Por exemplo, se um desenvolvedor corrigiu um problema em um serviço e obteve uma captura instantânea de seu aplicativo de processo ou de kit de ferramentas contido nesse ponto e, em seguida, um desenvolvedor diferente fez várias mudanças adicionais no mesmo serviço e obteve uma nova captura instantânea, o gerente de projeto poderá comparar as duas capturas instantâneas para determinar quais mudanças foram feitas quando e por quem. Se o gerente de projeto decidir que as mudanças adicionais no serviço não foram vantajosas, ele poderá reverter para a captura instantânea da correção original.

É possível executar diferentes versões (capturas instantâneas) de um aplicativo de processo simultaneamente em um servidor; durante a instalação de uma nova captura instantânea, remova a original ou deixe-a em execução.

Contexto da Versão

Cada captura instantânea possui metadados exclusivos para identificar a versão (referida como contexto da versão). Você designa o identificador, mas a IBM recomenda usar um sistema de versão numérica de três dígitos no formato <major>.<minor>.<service>. Consulte os tópicos sobre convenções de nomenclatura para obter uma descrição mais detalhada deste esquema de versão.

O IBM Business Process Manager designa um namespace global para cada aplicativo de processo. O namespace global é especificamente a dica do aplicativo de processo ou uma captura instantânea de aplicativo de processo específica. O nome da versão usado pelo servidor não pode ter mais de sete caracteres, portanto, o nome designado será um acrônimo que usa caracteres do nome de captura instantânea designado. Os acrônimos de captura instantânea serão idênticos aos seus nomes de captura instantânea se os nomes de captura instantânea estiverem em conformidade com o estilo IBM VRM recomendado e não tiverem mais de sete caracteres. Por exemplo, um nome de captura instantânea de 1.0.0 terá um acrônimo de 1.0.0 e um nome de captura instantânea de 10.3.0 terá o acrônimo de 10.3.0. O acrônimo de captura instantânea terá garantia de ser exclusivo no contexto do aplicativo de processo dentro do escopo do servidor Process Center. Por essa razão, não é possível editar o acrônimo da captura instantânea.

Considerações sobre Versão para Aplicativos de Processo em Diversos Clusters

É possível instalar a mesma versão de um aplicativo de processo em diversos clusters dentro da mesma célula. Para diferenciar entre essas diversas instalações da mesma versão do aplicativo de processo, crie uma captura instantânea para cada instalação e inclua um ID de célula exclusivo no nome da captura instantânea (por exemplo, v1.0_cell1_1 e v1.0_cell1_2). Cada captura instantânea é uma nova versão do aplicativo de processo (de uma perspectiva pura de gerenciamento de ciclo de vida), mas o conteúdo e a função são os mesmos.

Quando você instala um aplicativo de processo em um cluster, uma sincronização automática dos nós é executada.

Considerações sobre Versão para Kits de Ferramentas do Process Designer

Lembre-se de que capturas instantâneas de aplicativo de processo são geralmente obtidas quando se está pronto para testar ou instalar. No entanto, capturas instantâneas de kits de ferramentas são geralmente obtidas quando se está pronto para que esse kit de ferramentas seja usado pelos aplicativos de processo. Posteriormente, se desejar atualizar o kit de ferramentas, você deverá obter uma outra captura instantânea de "dica" quando estiver pronto e, então, os proprietários de aplicativos de processo e kits de ferramentas poderão decidir se desejam mover para cima para a nova captura instantânea.

Criando Versão de Módulos e Bibliotecas

Se um módulo ou biblioteca estiver em um aplicativo de processo ou kit de ferramentas, ele(a) assumirá o ciclo de vida do aplicativo de processo ou kit de ferramentas (versões, capturas instantâneas, rastreios e assim por diante). Nomes de módulo e de biblioteca devem ser exclusivos dentro do escopo de um aplicativo de processo ou kit de ferramentas.

Este tópico descreve a versão de módulos e bibliotecas que são usados com aplicativos de processo. Observe, no entanto, que, se você implementar módulos diretamente a partir do IBM Integration Designer para o Process Server, será possível continuar seguindo o procedimento de designação de números da versão para módulos durante a implementação, conforme descrito em "Criando Módulos e Bibliotecas Com Versão".

Um módulo ou uma biblioteca que está associada ao IBM Process Center deve ter suas bibliotecas dependentes no mesmo aplicativo de processo ou em um kit de ferramentas dependente.

A tabela a seguir lista as seleções que podem ser feitas no editor de dependência no IBM Integration Designer quando uma biblioteca está associada a um aplicativo de processo ou kit de ferramentas:

Tabela 36. Dependências para Módulo, Aplicativo de Processo ou Kit de Ferramentas e Bibliotecas Globais

Escopo da Biblioteca	Descrição	Pode depender de . . .
Módulos	Uma cópia desta biblioteca existe no servidor para cada módulo que ele usa.	Uma biblioteca com escopo definido no módulo pode depender de todos os tipos de bibliotecas.
Aplicativo de Processo ou Kit de Ferramentas	A biblioteca é compartilhada entre todos os módulos no escopo do aplicativo de processo ou kit de ferramentas. Esta configuração entra em vigor se a implementação é feita através do IBM Process Center. Se a implementação ocorre fora do IBM Process Center, a biblioteca é copiada em cada módulo. Nota: As bibliotecas criadas no IBM Integration Designer versão 8 possuem um nível de compartilhamento de Aplicativo de Processo ou Kit de Ferramentas por padrão.	Uma biblioteca deste tipo pode depender somente de bibliotecas globais.
Global	A biblioteca é compartilhada entre todos os módulos que estão em execução.	Uma biblioteca global pode depender apenas de outras bibliotecas globais. Nota: Você deve configurar uma biblioteca compartilhada do WebSphere a fim de implementar a biblioteca global. Consulte “Dependências de Módulo e Biblioteca” para obter informações adicionais.

Módulos e Bibliotecas Associados com Aplicativos de Processo ou Kits de Ferramentas

Não é necessário criar versão de módulos e bibliotecas associados com aplicativos de processo ou kits de ferramentas.

Os módulos e as bibliotecas que são associados a um aplicativo de processo ou kit de ferramentas não precisam ter versão. Na verdade, não é possível criar uma versão de um módulo ou de uma biblioteca associada a um aplicativo de processo ou kit de ferramentas no editor de dependência. Módulos e bibliotecas associados a um aplicativo de processo ou kit de ferramentas usam capturas instantâneas, uma função no Process Center, para obter o mesmo resultado que uma versão.

As bibliotecas associadas a um aplicativo de processo ou kit de ferramentas não terão um número de versão requerido na seção Bibliotecas do editor de dependência porque nenhuma versão é necessária.

Convenções de Nomenclatura

Uma convenção de nomenclatura é usada para diferenciar as várias versões de um aplicativo de processo conforme ele se move pelo ciclo de vida de atualização, implementação, coimplementação, remoção de implementação e arquivamento.

Esta seção fornece as convenções que são usadas para identificar exclusivamente as versões de um aplicativo de processo.

Um *contexto de versão* é uma combinação de acrônimos que descrevem exclusivamente um aplicativo de processo ou kit de ferramentas. Cada tipo de acrônimo possui uma convenção de nomenclatura. O acrônimo é limitado a um comprimento máximo de sete caracteres do conjunto de caracteres [A-Z0-9_], exceto para o acrônimo de captura instantânea, que também pode incluir um ponto.

- O acrônimo do aplicativo de processo é criado quando o aplicativo de processo é criado. Ele pode ter um máximo de sete caracteres de comprimento.
- O acrônimo de captura instantânea é criado automaticamente quando a captura instantânea é criada. Ele pode ter um máximo de sete caracteres de comprimento.

Se o nome da captura instantânea atender os critérios para um acrônimo de captura instantânea válido, o nome da captura instantânea e o acrônimo serão iguais.

Nota: Ao usar a função de roteamento reconhecida pela versão do componente do fluxo de mediação, nomeie sua captura instantânea para que ela esteja em conformidade com o esquema `<version>.<release>.<modification>` (por exemplo, `1.0.0`). Como o acrônimo de captura instantânea é limitado a sete caracteres, os valores de dígitos são limitados a um máximo de cinco dígitos no total (cinco dígitos mais dois pontos). Portanto, tome cuidado quando os campos de dígitos forem incrementados, porque qualquer coisa além dos primeiros sete caracteres será truncada.

Por exemplo, um nome de captura instantânea `11.22.33` resulta em um acrônimo de captura instantânea `11.22.3`.

- O acrônimo de faixa é gerado automaticamente a partir do primeiro caractere de cada palavra do nome da faixa. Por exemplo, uma nova faixa criada com o nome `My New Track` resultaria em um valor de acrônimo igual a `MNT`.

O nome da faixa padrão e o acrônimo são **Main**. A implementação em um servidor IBM Process Center inclui o acrônimo de faixa no contexto de versão se o acrônimo de faixa não é **Main**.

Uma definição de processo de negócios em um aplicativo de processo é geralmente identificada pelo acrônimo do nome do aplicativo de processo, o acrônimo da captura instantânea e o nome da definição de processo de negócios. Escolha nomes exclusivos para as suas definições de processo de negócios sempre que possível. Quando existirem nomes duplicados, você poderá encontrar os seguintes problemas:

- Você poderá não conseguir expor as definições de processo de negócios como serviços da Web sem alguma forma de mediação.
- Você poderá não conseguir chamar uma definição de processo de negócios criada no IBM Process Designer a partir de um processo BPEL criado no IBM Integration Designer.

O contexto da versão varia, dependendo de como o aplicativo de processo é implementado.

Convenções de Nomenclatura para Implementações do Servidor do Process Center

No servidor IBM Process Center, é possível implementar uma captura instantânea de um aplicativo de processo bem como uma captura instantânea de um kit de ferramentas. Além disso, é possível implementar a dica de um aplicativo de processo ou a dica de um kit de ferramentas. (Uma *dica* é a versão de trabalho atual de seu aplicativo de processo ou kit de ferramentas.) O contexto de versão varia, dependendo do tipo de implementação.

Para os aplicativos de processo, a dica de aplicativo de processo ou a captura instantânea do aplicativo de processo específica é usada para identificar exclusivamente a versão.

Os kits de ferramentas podem ser implementados com um ou mais aplicativos de processo, mas o ciclo de vida de cada kit de ferramentas é ligado ao ciclo de vida do aplicativo de processo. Cada aplicativo de processo possui sua própria cópia do kit de ferramentas dependente ou kits de ferramentas implementados no servidor. Um kit de ferramentas implementado não é compartilhado entre aplicativos de processo.

Se a faixa associada ao aplicativo de processo for denominada como algo diferente do padrão de **Main**, o acrônimo da faixa também fará parte do contexto da versão.

Para obter informações adicionais, consulte a seção “Exemplos” na página 34, posteriormente neste tópico.

Capturas Instantâneas do Aplicativo de Processo

Para implementações de captura instantânea do aplicativo de processo, o contexto da versão é uma combinação dos seguintes itens:

- Acrônimo do nome do aplicativo de processo
- Acrônimo da faixa do aplicativo de processo (se uma faixa diferente de **Main** for usada)
- Acrônimo da captura instantânea do aplicativo de processo

Kits de Ferramentas Independentes

Para implementações de captura instantânea do kit de ferramentas, o contexto da versão é uma combinação dos itens a seguir:

- Acrônimo do nome do kit de ferramentas
- Acrônimo da faixa do kit de ferramentas (se uma faixa diferente de **Main** for usada)
- Acrônimo da Captura Instantânea do Kit de Ferramentas

Dicas

Dicas do aplicativo de processo são usadas durante o teste iterativo no Process Designer. Elas podem ser implementadas apenas nos servidores Process Center.

Para implementações da dica do aplicativo de processo, o contexto da versão é uma combinação dos itens a seguir:

- Acrônimo do nome do aplicativo de processo
- Acrônimo da faixa do aplicativo de processo (se uma faixa diferente de **Main** for usada)
- "Dica"

As dicas do kit de ferramentas também são usadas durante o teste iterativo no Process Designer. Elas não são implementadas em um servidor de produção.

Para implementações de dica do kit de ferramentas, o contexto da versão é uma combinação dos itens a seguir:

- Acrônimo do nome do kit de ferramentas
- Acrônimo da faixa do kit de ferramentas (se uma faixa diferente de **Main** for usada)
- "Dica"

Exemplos

Os recursos devem ser nomeados e identificados exclusivamente externamente usando o contexto da versão.

- A tabela a seguir mostra um exemplo de nomes que são identificados exclusivamente. Neste exemplo, uma dica de aplicativo de processo usa o nome de faixa padrão (**Main**):

Tabela 37. Dica do Aplicativo de Processo com o Nome de Faixa Padrão

Tipo de nome	Exemplo
Nome do Aplicativo de Processo	Aplicativo de Processo 1
Acrônimo do nome do aplicativo de processo	PA1
Faixa do aplicativo de processo	Principal
Acrônimo da faixa do aplicativo de processo	"" (quando a faixa é Main)
Captura instantânea do aplicativo de processo	

Tabela 37. Dica do Aplicativo de Processo com o Nome de Faixa Padrão (continuação)

Tipo de nome	Exemplo
Acrônimo da captura instantânea do aplicativo de processo	Dica

Quaisquer módulos SCA associados a esta dica de aplicativo de processo incluem o contexto de versão, conforme mostrado na tabela a seguir:

Tabela 38. Módulos SCA e Arquivos EAR Reconhecidos pela Versão

Nome do módulo SCA	Nome Reconhecido pela Versão	Nome de EAR/Aplicativo Reconhecido pela Versão
M1	PA1-Tip-M1	PA1-Tip-M1.ear
M2	PA1-Tip-M2	PA1-Tip-M2.ear

- A tabela a seguir mostra um exemplo de uma dica de aplicativo de processo que usa um nome de faixa não padrão:

Tabela 39. Dica de Aplicativo de Processo com Nome de Faixa Não Padrão

Tipo de nome	Exemplo
Nome do Aplicativo de Processo	Aplicativo de Processo 1
Acrônimo do nome do aplicativo de processo	PA1
Faixa do aplicativo de processo	Track1
Acrônimo da faixa do aplicativo de processo	T1
Captura instantânea do aplicativo de processo	
Acrônimo da captura instantânea do aplicativo de processo	Dica

Quaisquer módulos SCA associados a esta dica de aplicativo de processo incluem o contexto de versão, conforme mostrado na tabela a seguir:

Tabela 40. Módulos SCA e Arquivos EAR Reconhecidos pela Versão

Nome do módulo SCA	Nome Reconhecido pela Versão	Nome de EAR/Aplicativo Reconhecido pela Versão
M1	PA1-T1-Tip-M1	PA1-T1-Tip-M1.ear
M2	PA1-T1-Tip-M2	PA1-T1-Tip-M2.ear

Convenções de nomenclatura semelhantes se aplicam às implementações avançadas de captura instantânea e de dica do Kit de Ferramentas. Elas também se aplicam às capturas instantâneas instaladas no Servidor de Processo.

- A tabela a seguir mostra um exemplo de nomes que são identificados exclusivamente. Nesse exemplo, uma captura instantânea do aplicativo de processo usa o nome de rastreo padrão (**Main**):

Tabela 41. Captura Instantânea do Aplicativo de Processo com Nome de Rastreo Padrão

Tipo de nome	Exemplo
Nome do Aplicativo de Processo	Aplicativo de Processo 1
Acrônimo do nome do aplicativo de processo	PA1
Faixa do aplicativo de processo	Principal
Acrônimo da faixa do aplicativo de processo	"" (quando a faixa é Main)
Captura instantânea do aplicativo de processo	Captura Instantânea do Processo V1

Tabela 41. Captura Instantânea do Aplicativo de Processo com Nome de Rastreo Padrão (continuação)

Tipo de nome	Exemplo
Acrônimo da captura instantânea do aplicativo de processo	PSV1

Quaisquer módulos SCA associados a essa captura instantânea de aplicativo do processo incluem o contexto de versão, conforme mostrado na tabela a seguir:

Tabela 42. Módulos SCA e Arquivos EAR Reconhecidos pela Versão

Nome do módulo SCA	Nome Reconhecido pela Versão	Nome de EAR/Aplicativo Reconhecido pela Versão
M1	PA1-PSV1-M1	PA1-PSV1-M1.ear
M2	PA1-PSV1-M2	PA1-PSV1-M2.ear

- A tabela a seguir mostra um exemplo de uma captura instantânea de aplicativo de processo que usa um nome de faixa não padrão:

Tabela 43. Captura instantânea do aplicativo de processo com nome de rastreo não padrão

Tipo de nome	Exemplo
Nome do Aplicativo de Processo	Aplicativo de Processo 1
Acrônimo do nome do aplicativo de processo	PA1
Faixa do aplicativo de processo	Track1
Acrônimo da faixa do aplicativo de processo	T1
Captura instantânea do aplicativo de processo	Captura Instantânea do Processo V1
Acrônimo da captura instantânea do aplicativo de processo	PSV1

Quaisquer módulos SCA associados a essa captura instantânea de aplicativo do processo incluem o contexto de versão, conforme mostrado na tabela a seguir:

Tabela 44. Módulos SCA e Arquivos EAR Reconhecidos pela Versão

Nome do módulo SCA	Nome Reconhecido pela Versão	Nome de EAR/Aplicativo Reconhecido pela Versão
M1	PA1-T1-PSV1-M1	PA1-T1-PSV1-M1.ear
M2	PA1-T1-PSV1-M2	PA1-T1-PSV1-M2.ear

Convenções de Nomenclatura para Implementações do Process Server

No Process Server, é possível implementar a captura instantânea de um aplicativo de processo. O acrônimo da captura instantânea do aplicativo de processo é usado para identificar exclusivamente a versão.

Para implementações de captura instantânea do aplicativo de processo, o contexto da versão é uma combinação dos seguintes itens:

- Acrônimo do nome do aplicativo de processo
- Acrônimo da captura instantânea do aplicativo de processo

Os recursos devem ser nomeados e identificados exclusivamente externamente usando o contexto da versão. A tabela a seguir mostra um exemplo de nomes que são identificados exclusivamente:

Tabela 45. Exemplo de Nomes e Acrônimos

Tipo de nome	Exemplo
Nome do Aplicativo de Processo	Aplicativo de Processo 1
Acrônimo do nome do aplicativo de processo	PA1
Captura instantânea do aplicativo de processo	1.0.0
Acrônimo da captura instantânea do aplicativo de processo	1.0.0

Um recurso, como um módulo ou uma biblioteca, tem o contexto de versão como parte de sua identificação.

A tabela a seguir mostra um exemplo de dois módulos e como os arquivos EAR associados incluem o contexto da versão:

Tabela 46. Módulos SCA e Arquivos EAR Reconhecidos pela Versão

Nome do módulo SCA	Nome Reconhecido pela Versão	Nome de EAR/Aplicativo Reconhecido pela Versão
M1	PA1-1.0.0-M1	PA1-1.0.0-M1.ear
M2	PA1-1.0.0-M2	PA1-1.0.0-M2.ear

A tabela a seguir mostra um exemplo de duas bibliotecas com escopo definido no aplicativo de processo e como os arquivos JAR associados incluem o contexto da versão:

Tabela 47. Bibliotecas com Escopo Definido no Aplicativo de Processo e Arquivos JAR Reconhecidos pela Versão

Nome da Biblioteca com Escopo Definido no Aplicativo de Processo SCA	Nome Reconhecido pela Versão	JAR Name Reconhecido pela Versão
Lib1	PA1-1.0.0-Lib1	PA1-1.0.0-Lib1.jar
Lib2	PA1-1.0.0-Lib2	PA1-1.0.0-Lib2.jar

Ligações Reconhecidas pela Versão

Os aplicativos de processo podem conter módulos SCA que incluem ligações de importação e exportação. Quando você implementar aplicativos conjuntamente, a ligação para cada versão do aplicativo deverá ser exclusiva. Algumas ligações são atualizadas automaticamente durante a implementação para assegurar a exclusividade entre versões. Em outros casos, você precisa atualizar a ligação após a implementação para garantir sua exclusividade.

Uma ligação *reconhecida pela versão* tem o escopo definido em uma versão específica de um aplicativo de processo, o que garante sua exclusividade entre aplicativos de processo. As seções a seguir descrevem as ligações que são atualizadas automaticamente para serem reconhecidas pela versão bem como quaisquer ações que você precisa executar no tempo de execução quando uma ligação não é reconhecida pela versão. Para obter informações sobre coisas a considerar quando você cria módulos, consulte “Considerações ao Usar Ligações”.

SCA

O destino de uma ligação de SCA é renomeado automaticamente para ser reconhecido pela versão durante a implementação se as ligações de importação e exportação do módulo são definidas no mesmo escopo do aplicativo de processo.

Se as ligações não estiverem definidas no mesmo escopo do aplicativo de processo, uma mensagem de informação será registrada. Você deve modificar a ligação de importação após a implementação para alterar o endereço de destino do terminal. É possível usar o console administrativo para alterar o endereço de destino do terminal.

Serviço da Web (JAX-WS ou JAX-RPC)

O endereço de destino de terminal de uma ligação de serviço da Web é automaticamente renomeado para ser reconhecido pela versão durante a implementação se todas as condições a seguir forem verdadeiras:

- Você seguiu as convenções de nomenclatura padrão para o endereço:
`http://ip:port/ModuleNameWeb/sca/ExportName`
- O endereço do terminal é SOAP/HTTP.
- As ligações de importação e exportação do módulo são definidas no mesmo escopo do aplicativo de processo.

Se estas condições não forem verdadeiras, uma mensagem de informação será registrada. A ação então executada dependerá de como você está implementando seu aplicativo de processo:

- Se estiver implementando conjuntamente seu aplicativo de processo, você deverá renomear manualmente a URL do terminal SOAP/HTTP ou a fila de destino SOAP/JMS para ser exclusiva entre as versões do aplicativo de processo. É possível usar o console administrativo após a implementação para alterar o endereço de destino do terminal.
- Se estiver implementando apenas uma única versão do aplicativo de processo, você poderá ignorar esta mensagem

Para implementação conjunta de captura instantânea de ligação de serviço da Web SOAP/JMS, a ação executada depende de como você está implementando seu aplicativo de processo:

- Se a importação e a exportação de destino estão no mesmo aplicativo de processo, execute as seguintes etapas antes de publicar o aplicativo de processo para o Process Center e criar a captura instantânea:
 1. Altere a URL do terminal da exportação. Certifique-se de que o destino e a connection factory sejam exclusivos.
 2. Altere a URL do terminal da importação para que seja igual à que você especificou para a exportação na etapa anterior.
- Se a importação e a exportação de destino estiverem em aplicativos de processo diferentes, execute as etapas a seguir:
 1. Altere a URL do terminal da exportação. Certifique-se de que o destino e a connection factory sejam exclusivos.
 2. Publique o aplicativo de processo no Process Center.
 3. Crie a captura instantânea.
 4. Implemente o aplicativo de processo no Process Server.
 5. Use o console administrativo do WebSphere para alterar a URL do terminal da importação correspondente para que seja igual à que você especificou para a exportação.

HTTP

O endereço da URL do terminal de uma ligação HTTP é renomeada automaticamente para ser reconhecida pela versão durante a implementação se todas as condições a seguir forem verdadeiras:

- Você seguiu as convenções de nomenclatura padrão para o endereço:
`http(s)://ip:port/ModuleNameWeb/contextPathInExport`
- As ligações de importação e exportação do módulo são definidas no mesmo escopo do aplicativo de processo.

Se estas condições não forem verdadeiras, uma mensagem de informação será registrada. A ação então executada dependerá de como você está implementando seu aplicativo de processo:

- Se estiver implementando conjuntamente seu aplicativo de processo, você deverá renomear manualmente a URL do terminal para ser exclusiva entre versões do aplicativo de processo. É possível usar o console administrativo após a implementação para alterar o endereço de destino do terminal.
- Se estiver implementando apenas uma única versão do aplicativo de processo, você poderá ignorar esta mensagem

JMS e JMS Genérico

Ligações JMS e JMS genérica geradas pelo sistema são automaticamente reconhecidas pela versão.

Nota: Para ligações JMS e JMS genéricas definidas pelo usuário, não ocorre nenhuma renomeação automática durante a implementação para ativar as ligações a serem reconhecidas pela versão. Se a ligação for definida pelo usuário, você deverá renomear os atributos a seguir para serem exclusivos entre versões do aplicativo de processo:

- Configuração de terminal
- Fila de destino de recebimento
- Nome da porta do listener (se definido)

Configure o destino de Envio correspondente se alterar o terminal do módulo de destino.

MQ/JMS e MQ

Nenhuma renomeação automática ocorre durante a implementação para permitir que ligações de tipo MQ/JMS ou MQ sejam reconhecidas pela versão.

Você deve renomear os seguintes atributos para serem exclusivos entre as versões do aplicativo de processo:

- Configuração de terminal
- Fila de destino de recebimento

Configure o destino de Envio correspondente se alterar o terminal do módulo de destino.

EJB

Não ocorre nenhuma renomeação automática durante a implementação para ativar ligações do tipo EJB para serem reconhecidas pela versão.

Você deve renomear o atributo de nome JNDI para ser exclusivo entre as versões do aplicativo de processo.

Observe que os aplicativos clientes também precisam ser atualizados para usar os novos nomes JNDI.

EIS

Um adaptador de recursos é automaticamente renomeado para ser reconhecido pela versão durante a implementação contanto que o nome do recurso padrão (*ModuleNameApp:Adapter Description*) não tenha sido modificado.

Se o nome do recurso padrão foi modificado, os nomes de adaptador de recursos deverão ser exclusivos entre as versões do aplicativo de processo.

Se os nomes de adaptador de recursos não forem exclusivos, uma mensagem informativa será registrada durante a implementação para alertá-lo. É possível renomear manualmente os adaptadores de recursos

após a implementação usando o console administrativo.

Chamada Dinâmica Reconhecida pela Versão

É possível configurar componentes do fluxo de mediação para rotear mensagens para terminais que são determinados dinamicamente no tempo de execução. Quando você cria o módulo de mediação, configura a consulta de terminal para usar roteamento reconhecido pela versão.

Se você usar o estilo IBM_VRM (*<version>.<release>.<modification>*) para a captura instantânea, poderá exportar o arquivo EAR do aplicativo de processo para o WebSphere Service Registry and Repository (WSRR). Quando você cria o módulo de mediação, em seguida, configura a consulta de terminal para usar roteamento reconhecido pela versão. Por exemplo, você seleciona **Retornar versão compatível mais recente correspondente ao terminal dos serviços baseados em módulo SCA** a partir do campo **Corresponder Política** e você seleciona **SCA** para o **Tipo de Ligação**.

Futuras versões do aplicativo de processo são implementadas no servidor e publicadas no WSRR e a consulta de terminal de módulo de mediação chama dinamicamente a versão compatível mais recente do terminal em serviço.

Observe que, como alternativa, você pode configurar o destino no SMOHeader e o valor pode ser transportado pela mensagem de solicitação.

Implementando Aplicativos de Processo com Módulos e Projetos Java

Aplicativos de processo podem conter módulos Java EE e projetos Java customizados. Quando você implementa aplicativos em conjunto, o módulo Java EE customizado para cada versão do aplicativo deve ser exclusivo.

Observe que os módulos Java EE e projetos Java customizados são implementados em um servidor se eles são implementados com um módulo SCA que possui uma dependência declarada deles. Se você não selecionar **Implementar com Módulo** (que é o padrão) quando declarar a dependência, será necessário implementar o módulo ou projeto manualmente.

Implementando Aplicativos de Processo com Regras de Negócios e Seletores

Se você estiver implementando diversas versões de um aplicativo de processo que inclui uma regra de negócios ou componente seletor, esteja ciente da maneira como os metadados associados são usados pelas versões.

Os metadados dinâmicos para uma regra de negócios ou componente seletor são definidos no tempo de execução pelo nome do componente, namespace de destino do componente e o tipo de componente. Se duas ou mais versões de um aplicativo de processo que contém uma regra de negócios ou seletor forem implementadas no mesmo ambiente de tempo de execução, elas compartilharão a mesma lógica de regra (regra de negócios) ou metadados de roteamento (seletor).

Para permitir que cada versão da regra de negócios ou componente seletor do aplicativo de processo use seus próprios metadados dinâmicos (lógica da regra ou roteamento), refatore o namespace de destino para que seja exclusivo para cada versão do aplicativo de processo.

Objetos de Configuração

É possível usar a ferramenta de administração de linha de comandos AdminConfig (wsadmin) do WebSphere para acessar e modificar as propriedades do banco de dados e de segurança no IBM Business Process Manager.

O termo *objeto de configuração* se refere a um objeto que é acessado usando os comandos wsadmin AdminConfig. Para obter mais informações, consulte Propriedades de Configuração de Segurança.

Arquitetura de Implementação

A arquitetura de implementação do IBM Business Process Manager consiste em processos de software chamados servidores, unidades topológicas referenciadas como nós e células, e o repositório de configuração usado para armazenar as informações de configuração.

Células

No IBM Business Process Manager, *Células* são agrupamentos lógicos de um ou mais nós em uma rede distribuída.

Uma célula é um conceito de configuração, uma forma de os administradores associar logicamente nós entre si. Os administradores definem os nós que compõem uma célula, de acordo com critérios específicos que fazem sentido em seus ambientes organizacionais.

Dados da configuração administrativa são armazenados em arquivos XML. Uma célula retém arquivos de configuração principal para cada servidor em cada nó na célula. Cada nó e servidor também têm seus próprios arquivos de configuração local. As alterações em um nó local ou em um arquivo de configuração de servidor serão temporárias, se o servidor pertencer à célula. Enquanto em efeito, as alterações locais substituem as configurações da célula. Alterações no servidor principal e nos arquivos de configuração do nó principal feitas no nível de célula substituem qualquer alteração temporária feita no nó quando os documentos de configuração da célula estão sincronizados nos nós. A sincronização ocorre em eventos designados, tal como quando um servidor inicia.

Servidores

Servidores fornecem a principal funcionalidade do IBM Business Process Manager. Os servidores de processo estendem ou aumentam a possibilidade de um servidor de aplicativos em manipular os módulos da Service Component Architecture (SCA). Outros servidores (gerenciadores de implementação e agentes de nó) são utilizados para gerenciar servidores de processo.

Um servidor de processo pode ser um *servidor independente* ou um *servidor gerenciado*. Um servidor gerenciado pode, opcionalmente, ser um membro de um *cluster*. Uma coleta de servidores gerenciados, clusters de servidores e outros tipos de middleware é chamada de *ambiente de implementação*. Em um ambiente de implementação, cada um dos servidores ou clusters gerenciados é configurado para uma função específica no ambiente de implementação (por exemplo, host de destino, host do módulo do aplicativo ou o servidor da Common Event Infrastructure). Um servidor independente é configurado para fornecer todas as funções necessárias.

Os servidores fornecem o ambiente de tempo de execução para os módulos SCA, para os recursos que são usados por esses módulos (origens de dados, especificações de ativação e destinos JMS), e para recursos fornecidos pela IBM (destinos de mensagens, contêineres do Business Process Choreographer e servidores da Common Event Infrastructure).

Um *agente de nó* é um agente administrativo que representa um nó para o seu sistema e gerencia os servidores nesse nó. Os agentes de nós monitoram os servidores em um sistema host e roteiam pedidos administrativos para os servidores. O agente do nó é criado quando um nó é associado a um gerenciador de implementação.

Um *gerenciador de implementação* é um agente administrativo que fornece uma visualização de gerenciamento centralizada para vários servidores e clusters.

Um servidor independente é definido pelo perfil independente; um gerenciador de implementação é definido por um perfil de gerenciador de implementação; servidores gerenciados são criados em um *nó gerenciado*, que está definido por um perfil gerenciado por nó.

Servidores Independentes

Um servidor independente fornece um ambiente para implementar módulos de SCA em um processo do servidor. Este processo do servidor inclui, mas não se limita a, um console administrativo, um destino de implementação, o suporte ao sistema de mensagens, o gerenciador de regras de processo de negócios e um servidor de Common Event Infrastructure.

Um servidor independente é simples de configurar e possui um console de iniciação rápida do qual é possível iniciar e parar o servidor e abrir a galeria de amostras e o console administrativo. Se você instalar as amostras do IBM Business Process Manager e, em seguida, abrir a galeria de amostras, uma solução de amostra será implementada no servidor independente. Você pode explorar os recursos utilizados para esta amostra no console administrativo.

É possível implementar suas próprias soluções em um servidor independente, mas um servidor independente não pode fornecer a capacidade, a escalabilidade ou a robustez requeridas de um ambiente de produção. Para o seu ambiente de produção, é melhor utilizar um ambiente de implementação de rede.

É possível iniciar com um servidor independente e posteriormente incluí-lo em um ambiente de implementação de rede, federando-o para uma célula do gerenciador de implementação, *contanto que nenhum outro nó tenha sido federado nessa célula*. Não é possível federar vários servidores independentes em uma célula. Para federar o servidor independente, utilize o console administrativo do gerenciador de implementação ou o comando **addNode**. O servidor independente não deve estar em execução ao federá-lo utilizando o comando **addNode**.

Um servidor independente é definido por um perfil do servidor independente.

Clusters

Os clusters são grupos de servidores que são gerenciados juntos e participam do gerenciamento de carga de trabalho.

Um cluster pode conter nós ou servidores de aplicativos individuais. Um nó geralmente é um sistema de computador físico com um endereço IP de host distinto que está executando um ou mais servidores de aplicativos. Clusters podem ser agrupados sob a configuração de uma célula, que associa logicamente muitos servidores e clusters com diferentes configurações e aplicativos um com o outro dependendo do critério do administrador e do que faz sentido em seus ambientes organizacionais.

Os clusters são responsáveis pelo equilíbrio de carga de trabalho entre os servidores. Servidores que são uma parte de um cluster são chamados de membros de cluster. Quando você instala um aplicativo em um cluster, o aplicativo é instalado automaticamente em cada membro de cluster.

Como cada membro de cluster contém os mesmos aplicativos, é possível distribuir tarefas do cliente em plataformas distribuídas de acordo com as capacidades das diferentes máquinas designando pesos a cada servidor.

Designar pesos aos servidores em um cluster melhora o desempenho e o failover. As tarefas são designadas aos servidores que têm a capacidade de executar as operações de tarefas. Se um servidor estiver disponível para executar a tarefa, ela será designada para outro membro do cluster. Este recurso de redesignação possui vantagens óbvias sobre a execução de um único servidor de aplicativos que pode se tornar sobrecarregado se muitas solicitações são feitas.

Perfis

Um perfil define um ambiente de tempo de execução exclusivo, com arquivos de comandos, arquivos de configuração e arquivos de log separados. Os perfis definem três tipos diferentes de ambientes em sistemas IBM Business Process Manager: servidor independente, gerenciador de implementação e nó gerenciado.

Ao usar perfis, é possível ter mais de um ambiente de tempo de execução em um sistema, sem ter de instalar cópias múltiplas dos arquivos binários do IBM Business Process Manager.

Use o Utilitário de Linha de Comandos BPMConfig para criar perfis do IBM BPM. O utilitário de linha de comandos **manageprofiles** ou sua interface gráfica com o usuário, o Profile Management Tool (PMT), pode ser usado como formas alternativas de criar o gerenciador de implementação ou perfis gerenciados por nós. O PMT não é mais suportado para criação de perfil independente.

Nota: Em plataformas distribuídas, cada perfil possui um nome exclusivo. No z/OS, todos os perfis são denominados “padrão”; não é possível renomear, editar, copiar ou excluir perfis no z/OS.

Tipos de Perfil

Os tipos de perfil a seguir do IBM BPM estão disponíveis com o IBM Business Process Manager V8.5:

Perfil independente do IBM BPM

O perfil independente define servidores independentes com recursos e funcionalidade específicos às configurações do IBM BPM Express. É possível criar o perfil independente usando o modelo de perfil BPM/BpmServer, que é suportado somente no IBM BPM Express.

Perfil de gerenciador de implementação do IBM BPM

O perfil de gerenciador de implementação define um gerenciador de implementação que fornece uma interface administrativa para um grupo lógico de servidores em uma ou mais estações de trabalho. É possível criar o perfil de gerenciador de implementação usando o modelo de perfil BPM/BpmDmgr, que é suportado somente no IBM BPM Standard e IBM BPM Advanced.

Perfil gerenciado por nó do IBM BPM

O perfil gerenciado por nó define um nó gerenciado quando o nó é federado para um gerenciador de implementação. É possível criar o perfil gerenciado por nó usando o modelo de perfil BPM/BpmNode, que é suportado somente no IBM BPM Standard e IBM BPM Advanced.

Os tipos de perfil específicos estão disponíveis com cada configuração do IBM BPM.

Tabela 48. Tipos de Perfil Disponíveis do IBM BPM

Configuração do IBM BPM	Tipos de Perfil		
	Independente	Gerenciador de Implementação	Gerenciado por nó
IBM BPMExpresso	Sim	No	No
IBM BPM Standard	No	Sim	Sim
IBM BPM Advanced	No	Sim	Sim
Unit Test Environment (UTE) for IBM Integration Designer	Sim	Opcional	Opcional

O Diretório do Perfil

Cada perfil no sistema tem seu próprio diretório, contendo todos seus arquivos. O local do diretório de perfil é especificado ao criar o perfil. Por padrão, ele se encontra no diretório `profiles`, no diretório no qual o IBM Business Process Manager está instalado. Por exemplo, o perfil Dmgr encontra-se em `C:\Program Files\IBM\WebSphere\ProcServer\profiles\Dmgr`.

O Perfil Padrão

O primeiro perfil criado dentro de uma instalação do IBM Business Process Manager é o *perfil padrão*. O perfil padrão é o destino padrão para comandos emitidos a partir do diretório `bin` no diretório em que o IBM Business Process Manager foi instalado. Se existir apenas um perfil em um sistema, todos os comandos operarão nesse perfil. Se você criar um outro perfil, você poderá torná-lo o padrão.

Nota: O perfil padrão não é necessariamente um perfil cujo nome é “default”.

Aprimorando Perfis

Se já tiver um perfil de gerenciador de implementação, um perfil gerenciado por nó ou um perfil do servidor independente criado para WebSphere Application Server Network Deployment, é possível *aumentá-lo* para suportar o IBM Business Process Manager além da função existente. Para aprimorar um perfil, primeiro instale o IBM Business Process Manager. Em seguida, use o utilitário de linha de comandos **manageprofiles** para aumentar um perfil independente ou use o Profile Management Tool ou o utilitário de linha de comandos **manageprofiles** para aumentar um gerenciador de implementação ou um perfil gerenciado por nó.

Importante: Em um ambiente de implementação de rede, você precisa aumentar o perfil de gerenciador de implementação primeiro e depois aumentar os perfis de nó gerenciado.

Restrição: Não é possível aumentar um perfil de gerenciador independente ou de implementação onde o registro do usuário do WebSphere VMM padrão foi alterado, por exemplo, para usar o LDAP.

Gerenciadores de implementação

Um gerenciador de implementação é um servidor que gerencia operações para um grupo lógico, ou célula, de outros servidores. O gerenciador de implementação é o local central para administração de servidores e clusters.

Ao criar um ambiente de implementação, o perfil do gerenciador de implementação é o primeiro perfil a ser criado. Cada ambiente de implementação criado possui um console de iniciação rápida, a partir do qual é possível iniciar e parar o gerenciador de implementação e iniciar seu console administrativo. Utilize o console administrativo do gerenciador de implementação para gerenciar os servidores e clusters na célula. Isso inclui configurar servidores e clusters, incluir servidores nos clusters, iniciar e parar servidores e clusters e implementar os módulos SCA.

Embora o gerenciador de implementação seja um tipo de servidor, não é possível implementar módulos no próprio gerenciador de implementação.

Nós

Um *nó* é um agrupamento lógico de servidores gerenciados.

Um nó geralmente corresponde a um sistema de computador lógico ou físico com um endereço de host IP diferente. Os nós não podem abranger múltiplos computadores. Os nós de nomes geralmente são idênticos ao nome do host do computador.

Os nós na topologia de implementação de rede podem ser gerenciados ou não gerenciados. Um nó gerenciado tem um processo de agente do nó que gerencia sua configuração e servidores. Os nós não gerenciados não têm um agente do nó.

Nós Gerenciados

Um *nó gerenciado* é um nó que está associado a um gerenciador de implementação e que contém um agente do nó, podendo conter servidores gerenciados. Em um nó gerenciado, você pode configurar e executar servidores gerenciados.

Os servidores que são configurados em um nó gerenciado, formam os recursos de seu ambiente de implementação. Estes servidores são criados, configurados, iniciados, parados, gerenciados e excluídos utilizando o console administrativo do gerenciador de implementação.

Um nó gerenciado contém um agente do nó que gerencia todos os servidores em um nó.

Quando um nó é federado, um processo do agente do nó é criado automaticamente. Este agente do nó deve estar em execução para ser capaz de gerenciar a configuração do perfil. Por exemplo, quando você realiza as seguintes tarefas:

- Iniciar e parar processos do servidor.
- Sincronizar dados de configuração no gerenciador de implementação com a cópia no nó.

Entretanto, o agente do nó não precisa estar em execução para que os aplicativos sejam executados ou para configurar recursos no nó.

Um nó gerenciado pode conter um ou mais servidores, que são gerenciados por um gerenciador de implementação. É possível implementar soluções para os servidores em um nó gerenciado, mas o nó gerenciado não contém uma galeria de aplicativos de amostras. O nó gerenciado é definido por um perfil gerenciado por nó e possui um console de Iniciação Rápida.

Nós não gerenciados

Um nó não gerenciado não tem um agente de nó para gerenciar seus servidores.

Os nós não gerenciados na topologia de Implementação de Rede podem ter definições de servidor, como servidores da web, mas não definições de servidor de aplicativos. Os nós não gerenciados nunca podem ser federados. Ou seja, um agente do nó nunca pode ser incluído em um nó não gerenciado. Um outro tipo de nó não gerenciado é um servidor independente. O gerenciador de implementação não pode gerenciar este servidor independente porque ele não é conhecido na célula. Um servidor independente pode ser federado. Quando ele é federado, um agente do nó é criado automaticamente. O nó se torna um nó gerenciado na célula.

Agentes do nó

Os agentes do nó são agentes administrativos que roteiam pedidos administrativos para servidores.

Um agente do nó é um servidor que é executado em cada sistema do computador host que participa na configuração do Network Deployment. Ele é puramente um agente administrativo e não está envolvido nas funções de serviço do aplicativo. Um agente do nó também hospeda outras funções administrativas importantes tais como serviços de transferência de arquivos, sincronização de configuração e monitoramento de desempenho.

Considerações sobre a Nomenclatura de Perfis, Nós, Servidores, Hosts e Células

Este tópico discute os problemas e os termos reservados que deverão ser considerados ao nomear o perfil, o nó, o servidor, o host e a célula (se aplicável). Este tópico se aplica apenas às plataformas distribuídas.

Considerações de Nomenclatura do Perfil

O nome do perfil pode ser qualquer nome exclusivo com as restrições a seguir. Não utilize nenhum dos caracteres a seguir ao nomear seu perfil:

- Espaços
- Caracteres especiais que não são permitidos no nome de um diretório do sistema operacional, tal como *, & ou ?.

- Barras (/) ou barras invertidas (\)

Caracteres de byte duplo são permitidos.

Considerações sobre o caminho do diretório: O caminho do diretório de instalação deve ser menor ou igual a 60 caracteres. O número de caracteres no diretório *profiles_directory_path\profile_name* deve ser menor ou igual a 80 caracteres.

Nota: Use uma convenção de nomenclatura de caminho curto ao criar um perfil em um ambiente Windows para evitar a limitação de comprimento de caminho de 255 caracteres do Windows.

Considerações sobre Nomenclatura de Nó, Servidor, Host e Célula

Nomes reservados: Evite usar nomes reservados como valores de campos. O uso de nomes reservados pode causar resultados imprevisíveis. As seguintes palavras são reservadas:

- cells
- nodes
- servers
- clusters
- aplicativos
- deployments

Descrições dos campos nas páginas Nó e Nomes de Host e Nó, Host e Nomes de Célula: Use as diretrizes de nomenclatura apropriadas ao criar perfis.

- Perfis de servidores independentes
- Perfis do Gerenciador de Implementação
- Perfis gerenciados por nó

Tabela 49. Diretrizes de Nomenclatura para Perfis de Servidor Independente

Nome do Campo	Valor Padrão	Restrições	Descrição
Nome do nó	... <i>shortHostName</i> Node <i>NodeNumber</i> em que: <ul style="list-style-type: none"> • <i>shortHost Name</i> é o nome abreviado do host. • <i>NodeNumber</i> é um número sequencial que começa em 01. 	Evite utilizar os nomes reservados.	Selecione qualquer nome desejado. Para ajudar a organizar sua instalação, utilize um nome exclusivo se você planeja criar mais de um servidor no sistema.
Nome do servidor	... server1	Use um nome exclusivo para o servidor.	O nome lógico para o servidor.
Nome do host	... O formato longo do nome do DNS (Servidor de Nomes de Domínio).	Use um nome completo do host que seja endereçável por meio da sua rede.	Utilize o nome de DNS real ou o endereço IP de sua estação de trabalho para ativar a comunicação com ele. Consulte informações adicionais sobre o nome do host após esta tabela.

Tabela 50. Diretrizes de Nomenclatura para Perfis de Gerenciador de Implementação

Nome do Campo	Valor Padrão	Restrições	Descrição
Nome do nó	<p>... <i>shortHostName</i> Cell <i>ManagerNode Number</i> em que:</p> <ul style="list-style-type: none"> • <i>shortHost Name</i> é o nome abreviado do host. • <i>NodeNumber</i> é um número sequencial que começa em 01. 	<p>Utilize um nome exclusivo para o gerenciador de implementação. Evite utilizar os nomes reservados.</p>	<p>O nome é utilizado para administração na célula do gerenciador de implementação.</p>
Nome do host	<p>... O formato longo do nome do DNS (Servidor de Nomes de Domínio).</p>	<p>Use um nome completo do host que seja endereçável por meio da sua rede. Evite utilizar os nomes reservados.</p>	<p>Utilize o nome de DNS real ou o endereço IP de sua estação de trabalho para ativar a comunicação com ele. Consulte informações adicionais sobre o nome do host após esta tabela.</p>
Nome da Célula	<p>... <i>shortHostName</i> Cell <i>CellNumber</i> em que:</p> <ul style="list-style-type: none"> • <i>shortHost Name</i> é o nome abreviado do host. • <i>CellNumber</i> é um número sequencial que começa em 01. 	<p>Utilize um nome exclusivo para a célula do gerenciador de implementação. Um nome de célula deve ser exclusivo em qualquer circunstância na qual o produto está em execução na mesma estação de trabalho física ou cluster de estações de trabalho, como um Sysplex. Além disso, um nome de célula deve ser exclusivo em qualquer circunstância na qual a conectividade de rede entre entidades é requerida, entre as células ou a partir de um cliente que deve comunicar-se com cada uma das células. Os nomes de células também deverão ser exclusivos se seus espaços de nomes tiverem que ser associados. Caso contrário, você pode encontrar sintomas, tais como uma exceção <code>javax.naming.NameNotFoundException</code> que, nesse caso, é necessário criar células exclusivamente denominadas.</p>	<p>Todos os nós federados tornam-se membros da célula do gerenciador de implementação, que é nomeada na página Nomes de Nós, Hosts e Células do Profile Management Tool.</p>

Tabela 51. Diretrizes de Nomenclatura para Perfis Gerenciados por Nó

Nome do Campo	Valor Padrão	Restrições	Descrição
Nome do nó	... <i>shortHostName</i> Node <i>NodeNumber</i> em que: <ul style="list-style-type: none"> • <i>shortHost Name</i> é o nome abreviado do host. • <i>NodeNumber</i> é um número sequencial que começa em 01. 	Evite utilizar os nomes reservados. Utilize um nome exclusivo na célula do gerenciador de implementação.	O nome é usado para administração na célula do gerenciador de implementação à qual o perfil gerenciado por nó está incluída. Utilize um nome exclusivo na célula do gerenciador de implementação.
Nome do host	... O formato longo do nome do DNS (Servidor de Nomes de Domínio).	Use um nome completo do host que seja endereçável por meio da sua rede.	Utilize o nome de DNS real ou o endereço IP de sua estação de trabalho para ativar a comunicação com ele. Consulte informações adicionais sobre o nome do host após esta tabela.

Considerações sobre o Nome do Host:

O nome do host é o nome da rede para a estação de trabalho física na qual o nó está instalado. O nome do host deve indicar um nó de rede física no servidor. Quando há várias placas de rede no servidor, o nome do host ou o endereço IP deve indicar uma das placas de rede. Os nós remotos utilizam o nome do host para conexão e comunicação com o nó.

O IBM Business Process Manager é compatível com o Internet Protocol versão 4 (IPv4) e versão 6 (IPv6). Onde quer que você possa digitar endereços IP no console administrativo, ou em outro lugar qualquer, você pode fazê-lo em qualquer um dos formatos. Observe que, se o IPv6 estiver implementado em seu sistema, você deve digitar o endereço IP no formato IPv6, contudo, se o IPv6 ainda não estiver disponível, digite endereços IP no formato IPv4. Para obter informações adicionais sobre o IPv6 consulte a descrição a seguir : IPv6.

As seguintes diretrizes podem ajudar na determinação do nome de host apropriado para sua estação de trabalho:

- Selecione um nome de host que outras estações de trabalho possam acessar em sua rede.
- Não utilize o identificador genérico, host local, para este valor.
- Não tente instalar produtos do IBM Business Process Manager em um servidor com um nome do host que utiliza caracteres de DBCS (Conjunto de Caracteres de Byte Duplo). Caracteres DBCS não são suportados quando utilizados no nome do host.
- Evite utilizar o caractere sublinhado (_) nos nomes de servidor. Padrões da Internet declaram que nomes de domínio devem estar conformidade com os requisitos de nome do host descritos nos padrões Internet Official Protocol Standards RFC 952 e RFC 1123. Os nomes de domínio devem conter somente letras (maiúsculas ou minúsculas) e dígitos. Os nomes de domínio também podem conter caracteres de traço (-) contanto que os traços não estejam nas extremidades do nome. Os caracteres sublinhados (_) não são suportados no nome do host. Se você tiver instalado o IBM Business Process Manager em um servidor com um caractere sublinhado no nome do servidor, acesse o servidor com seu endereço IP até renomeá-lo.

Se você definir nós coexistentes no mesmo computador com endereços IP exclusivos, defina cada endereço IP em uma tabela de consulta de DNS (Domain Name Server). Os arquivos de configuração para servidores não fornecem a resolução de nome de domínio para endereços IP múltiplos em uma estação de trabalho com um único endereço de rede.

O valor que você especifica para o nome do host é utilizado como o valor da propriedade `hostName` nos documentos de configuração. Especifique o valor do nome do host em um dos seguintes formatos:

- Cadeia do nome completo do host DNS (Domain Name Servers), como xmachine.manhattan.ibm.com
- A cadeia do nome abreviado do host DNS padrão, como xmachine
- Endereço IP numérico, como 127.1.255.3

O nome completo do host DNS tem as vantagens de ser totalmente inequívoco e flexível. Você tem a flexibilidade de alterar o endereço IP real do sistema host sem ter de alterar a configuração do servidor. Esse valor para o nome do host é especialmente útil se você pretende alterar o endereço IP com frequência ao utilizar o DHCP (Dynamic Host Configuration Protocol) para atribuir endereços IP. Uma desvantagem desse formato é ser dependente do DNS. Se o DNS não estiver disponível, a conectividade ficará comprometida.

O nome abreviado do host também pode ser resolvido dinamicamente. Um formato de nome abreviado tem a capacidade adicional de ser redefinido no arquivo de hosts locais para que o sistema possa executar o servidor mesmo quando desconectado da rede. Defina o nome abreviado para 127.0.0.1 (auto-retorno local) nos arquivos de hosts para executar desconectado. Uma desvantagem do formato de nome abreviado é ser dependente do DNS para acesso remoto. Se o DNS não estiver disponível, a conectividade ficará comprometida.

Um endereço IP numérico tem a vantagem de não requerer uma resolução de nome por meio do DNS. Um nó remoto pode conectar-se ao nó denominado com um endereço IP numérico sem o DNS estar disponível. Uma desvantagem desse formato é o endereço IP numérico ser fixo. Você deve alterar a configuração da propriedade hostName nos documentos de configuração sempre que alterar o endereço IP da estação de trabalho. Portanto, não utilize um endereço IP numérico se você utilizar DHCP ou se você alterar os endereços IP regularmente. Outra desvantagem desse formato é não ser possível usar o nó se o host estiver desconectado da rede.

BPMN 2.0

As definições do processo de negócio do IBM Business Process Manager suportam a subclasse Executável Comum da classe de conformidade Modelagem de Processo do BPMN 2.0, que lida com modelos que podem ser executados.

O Business Process Model and Notation (BPMN) é o padrão fundamental para os processos no IBM Process Designer e no IBM Process Center. Os diagramas de Business Process Definition (BPD) baseiam-se na especificação do BPMN. Esse tópico apresenta algumas maneiras pelas quais o BPMN 2.0 é aplicado no IBM Business Process Manager. Para obter informações detalhadas sobre o BPMN, consulte a página de Especificação do BPMN em <http://www.bpmn.org/>.

O IBM Business Process Manager suporta os seguintes tipos de tarefas do BPMN 2.0:

- Nenhum (tarefa abstrata na especificação do BPMN 2.0)
- Tarefa do sistema (tarefa de serviço na especificação do BPMN 2.0)
- Tarefa do usuário
- Script
- Tarefa de decisão (tarefa de regras de negócio na especificação do BPMN 2.0)

Os eventos de mensagem intermediária do IBM BPM fornecem funções simulares à tarefa de envio e tarefa de recebimento do BPMN.

Notação do BPMN 2.0

A partir do V7.5.1, os ícones de tarefa do Process Designer BPMN 2.0 nos diagramas do BPD são coletados em uma paleta simplificada e exibidos nos diagramas de processo. Os ícones mostram se sua atividade é uma tarefa do sistema, tarefa do usuário, tarefa de decisão, script ou processo vinculado. As atividades em modelos que foram criados em versões anteriores também mostram tipos de tarefas e ícones de tarefas apropriadas do BPMN 2.0 ao visualizá-los na versão 7.5.1 ou posterior.

Atividades e Tarefas

Existem algumas mudanças de terminologia das versões anteriores do Process Designer. Algumas dessas mudanças envolvem tipos de atividade que foram renomeados.

- Atividades de serviço (automatizadas) agora são tarefas do sistema.
- Atividades de serviço (tarefa) em uma raia que não pertence ao sistema agora são tarefas do usuário.
- Atividades de serviço (tarefa) em uma raia do sistema agora são tarefas de decisão se fizerem referência a um serviço de decisão.
- Atividades de serviço (tarefa) em uma raia do sistema agora são tarefas do sistema se fizerem referência a qualquer tipo de serviço que não seja um serviço de decisão.
- Atividades Javascript agora são tarefas de script.
- Atividades de processo aninhadas agora são processos vinculados.
- Atividades externas de versões anteriores do Process Designer estão disponíveis como implementações externas para tarefas de usuários ou tarefas do sistema.

Gateways

Não há nenhuma mudança de notação para os gateways de versões anteriores. Entretanto, há três mudanças de terminologia. O gateway de decisão agora é *gateway exclusivo*, o gateway de divisão ou junção simples agora é *gateway paralelo*, e o gateway de divisão ou de junção condicional agora é *gateway inclusivo*.

Também há um novo tipo de gateway, o *gateway de evento*. Um gateway de evento representa um ponto de ramificação em um processo no qual os caminhos alternativos que seguem o gateway se baseiam nos eventos que ocorrem, em vez da avaliação das expressões usando dados do processo (assim como um gateway exclusivo ou inclusivo). Um evento específico, geralmente o recebimento de uma mensagem, determina o caminho que será usado.

Eventos de Não Interrupção

Notação incluída do BPMN 2.0 para eventos de não interrupção. Por padrão, um evento de limite interrompe a atividade à qual está anexada. Quando o evento é acionado, a atividade para e o token continua com o fluxo de sequência de saída do evento. Se o evento está configurado como não interrupção, quando o evento for acionado, a atividade anexada continua em paralelo e um novo token é gerado e passado para o fluxo de sequência de saída do evento. O limite de evento se altera para uma linha tracejada para os eventos de não interrupção.

Eventos intermediários que foram anexados em atividades são eventos intermediários de interrupção se fecharem suas atividades anexadas ou eventos intermediários de não interrupção se não fecharem suas atividades anexadas.

Iniciar evento

A especificação BPMN permite que os modelos de processo omitam símbolos de início e término de evento. O Process Designer requer que os modelos de processo usem eventos de início e término.

Há vários tipos de eventos de inícios disponíveis no Process Designer:

processos

- nenhum
- Mensagem
- ad hoc

subprocessos

- nenhum

subprocessos de evento

- erro
- Mensagem
- cronômetro

É possível alterar o tipo de um evento de início ao editar as propriedades do evento. É possível ter diversos eventos de início de mensagem em um processo, mas é possível usar somente um evento sem início.

Eventos de Término

Quatro tipos de eventos de terminos estão disponíveis: *mensagem*, *finalizar*, *erro* e *nenhum*. É possível alterar o tipo de um evento de término.

Quando um processo-pai chama um processo-filho e o processo-filho executa uma ação de evento de finalização, o processo-filho para e o processo-pai continua então com suas próximas etapas.

Subprocessos

A especificação BPMN define dois tipos de subprocessos, integrados e reutilizáveis. É possível criar ambos os tipos no Process Designer. Os subprocessos integrados são chamados apenas de *subprocessos* no Process Designer e na nova versão 7.5.1. O subprocesso reutilizável do BPMN é chamado de um *processo vinculado* no Process Designer.

Um subprocesso existe no processo contido e é uma forma de agrupar etapas de processo para reduzir a complexidade e a confusão do diagrama. Subprocessos reduzem diversas etapas a uma atividade. O subprocesso pode ser visto somente pelo processo no qual está definido. Um subprocesso existe no escopo de seu responsável pela chamada e possui acesso a todas as variáveis nesse ambiente. Nenhum parâmetro entra e sai do subprocesso integrado.

Separado do subprocesso e do processo vinculado, o Process Designer possui um subprocesso de evento, que é um subprocesso especializado que é usado para manipulação de eventos. Ele não é conectado as outras atividades por meio de fluxo de sequência, e ele ocorre somente se seu evento de início for acionado.

Processos Vinculados

Um subprocesso reutilizável do BPMN é chamado de *processo vinculado* no Process Designer. É um processo criado fora do processo atual que pode ser chamado pelo processo atual. Ele é reutilizável porque outras definições de processo também podem chamar esse processo. O processo vinculado define seus parâmetros de entrada e saída e não possui acesso ao escopo ou ambiente do responsável pela chamada. O processo vinculado é semelhante ao processo aninhado disponível em versões anteriores; não há mudança no comportamento da atividade. Processos aninhados anteriores são migrados para processos vinculados. O processo vinculado é semelhante a um subprocesso com um limite espesso e é destacado na janela do Inspector.

Loops

O BPMN fornece o conceito de uma atividade que pode ser repetida. A atividade pode ser atômica, o que significa que a atividade se repete, ou ela pode ser um subprocesso, encapsulando uma série de etapas que são repetidas. Se expandir a atividade repetida, verá as atividades contidas que devem ser executadas repetidamente. A condição é sempre avaliada no início de cada iteração de loop. Não há nenhuma possibilidade de avaliar o final de cada iteração de loop.

O IBM Business Process Manager possui um *loop multi-instância*, que é executado um número finito de vezes com as atividades contidas na execução sequencialmente ou em paralelo.

Importando Processos Não BPMN

É possível importar modelos que foram criados no IBM WebSphere Business Modeler e usá-los no Process Designer. Para obter informações sobre a importação do BPMN 2.0, consulte Mapeando elementos do IBM WebSphere Business Modeler para construções do IBM Business Process Manager. Também é possível importar modelos do BPMN 2.0 que foram criados no IBM WebSphere Business Compass, Rational Software Architect ou outros ambientes de modelagem.

Business Process Definitions (BPDs)

Para modelar um processo no IBM Process Designer, você deve primeiro criar uma Business Process Definition (BPD). A definição de processo de negócios pode basear-se em um modelo BPMN importado.

Uma Business Process Definition é um modelo reutilizável de um processo que define o que é comum a todas as instâncias de tempo de execução desse modelo de processo. Uma BPD deve conter um evento de início, um evento de término e pelo menos uma pista, e uma ou mais atividades. Consulte "Convenções de nomenclatura do Designer de Processo IBM" nos links relacionados para obter detalhes sobre as limitações de caractere que se aplicam às BPDs.

Uma Business Process Definition (BPD) precisa incluir uma pista para cada sistema ou grupo de usuários que participa em um processo. Uma pista pode ser uma pista participante ou uma pista do sistema. Porém, é possível criar uma BPD que agrupe as atividades de um grupo e um sistema em uma única pista se essa for a sua preferência. Consulte "Criando uma Business Process Definition (BPD)" nos links relacionados para informações sobre como criar uma BPD.

É possível designar qualquer pessoa ou grupo específico para ser responsável pelas atividades em uma pista participante. Cada pista criada é designada ao grupo Todos os Usuários por padrão. É possível usar este grupo padrão para execução e teste do seu BPD no Inspector. O grupo Todos os Usuários inclui todos os usuários que são membros do grupo de segurança **tw_allusers**, que é um grupo de segurança especial que inclui automaticamente todos os usuários no sistema.

Uma pista do sistema contém atividades manipuladas por um sistema IBM Process Center específico. Cada atividade precisa de uma implementação, que define a atividade e configura as propriedades para a tarefa. Durante a implementação, um desenvolvedor cria um serviço ou grava o JavaScript necessário para concluir as atividades em uma pista do sistema. Consulte "Entendendo tipos de serviço" nos links relacionados para informações sobre serviços.

Para cada BPD que criar, você precisará declarar variáveis para capturar os dados de negócios que são passados de atividade para atividade no seu processo. Consulte "Gerenciando e Mapeando Variáveis" nos links relacionados para aprender sobre implementação de variáveis.

Também é possível incluir eventos em uma BPD. Eventos em uma definição de processo de negócios IBM podem ser acionados pela passagem de um prazo final, uma exceção, ou a chegada de uma mensagem. O acionador que você deseja determina o tipo de evento que você escolhe para implementar. Para informações detalhadas sobre tipos de evento disponíveis e seus acionadores, consulte "Modelando eventos".

Ligações

No núcleo de uma arquitetura orientada a serviços está o conceito de um *serviço*, uma unidade de funcionalidade executada por uma interação entre dispositivos de computação. Uma *exportação* define a interface externa (ou ponto de acesso) de um módulo, de forma que os componentes Service Component Architecture (SCA) dentro do módulo possam fornecer seus serviços para clientes externos. Uma

importação define uma interface para serviços fora de um módulo, de forma que os serviços possam ser chamados a partir de dentro do módulo. Você pode usar *ligações* específicas do protocolo com importações e exportações para especificar os meios de transportar os dados para dentro ou para fora do módulo.

Exportações

Clientes externos podem chamar os componentes SCA em um módulo de integração por meio de vários protocolos (como HTTP, JMS, MQ e RMI/IIOP) com dados em uma variedade de formatos (como XML, CSV, COBOL e JavaBeans). As exportações são componentes que recebem essas solicitações de origens externas e, em seguida, chamam componentes do IBM Business Process Manager usando o modelo de programação SCA.

Por exemplo, na figura a seguir, uma exportação recebe uma solicitação sobre o protocolo HTTP a partir de um aplicativo cliente. Os dados são transformados em um objeto de negócios, o formato usado pelo componente SCA. O componente é, então, chamado com esse objeto de dados.

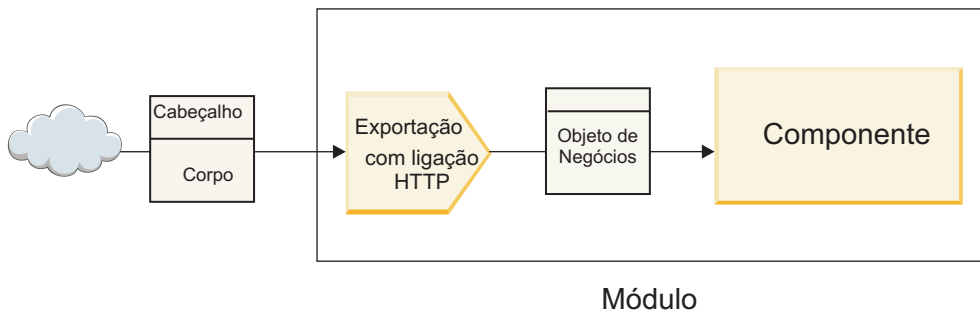


Figura 46. Uma Exportação com Ligação HTTP

Importações

Um componente SCA pode desejar chamar um serviço externo não SCA que espera dados em um formato diferente. Uma importação é usada pelo componente SCA para chamar o serviço externo usando o modelo de programação SCA. A importação, então, chama o serviço de destino da maneira que o serviço espera.

Por exemplo, na figura a seguir, uma solicitação a partir de um componente SCA é enviada, pela importação, para um serviço externo. O objeto de negócios, que é o formato usado pelo componente SCA, é transformado no formato esperado pelo serviço e o serviço é chamado.

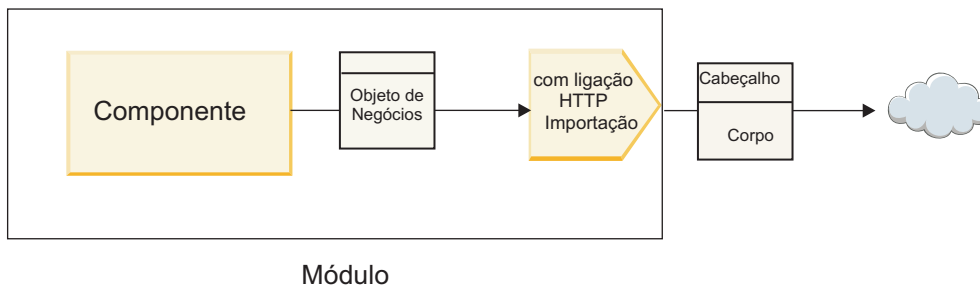


Figura 47. Uma Importação com Ligação HTTP

Lista de Ligações

Use o Integration Designer para gerar uma ligação para uma importação ou exportação e para configurar a ligação. Os tipos de ligações que estão disponíveis são descritos na lista a seguir.

- SCA

A ligação SCA, que é o padrão, permite que seu serviço se comunique com serviços em outros módulos SCA. É possível usar uma importação com uma ligação SCA para acessar um serviço em um outro módulo SCA. Use uma exportação com uma ligação SCA para oferecer um serviço para outros módulos SCA.

- Serviço da Web

Uma ligação de serviço da web permite acessar um serviço externo usando mensagens SOAP interoperáveis e qualidades de serviço. Também é possível usar ligações de serviço da web para incluir anexos como parte da mensagem SOAP.

A ligação de serviço da web pode usar um protocolo de transporte de SOAP/HTTP (SOAP sobre HTTP) ou SOAP/JMS (SOAP sobre JMS). Independentemente do transporte (HTTP ou JMS) usado para transmitir as mensagens SOAP, as ligações de serviço da web sempre manipulam interações de solicitação/resposta de forma síncrona.

- HTTP

A ligação HTTP permite que você acesse um serviço externo usando o protocolo HTTP, onde as mensagens não SOAP são usadas ou onde o acesso HTTP direto é necessário. Esta ligação é usada ao trabalhar com serviços da web baseados no modelo de HTTP (isto é, serviços que usam operações de interface HTTP bem conhecidas como GET, PUT, DELETE e assim por diante).

- Enterprise JavaBeans (EJB)

As ligações EJB permitem que componentes SCA interajam com os serviços fornecidos pela lógica de negócios Java EE em execução em um servidor Java EE.

- EIS

A ligação EIS (Enterprise Information System), quando usada com um adaptador de recursos JCA, permite acessar serviços em um Enterprise Information System ou tornar seus serviços disponíveis para o EIS.

- Ligações JMS

As ligações Java Message Service (JMS), JMS genéricas e JMS do WebSphere MQ (JMS do MQ) são usadas para interações com sistemas de mensagens, onde a comunicação assíncrona por meio das filas de mensagens é crítica para confiabilidade.

Uma exportação com uma das ligações JMS observa uma fila para a chegada de uma mensagem e envia de forma assíncrona a resposta, se houver, para a fila de resposta. Uma importação com uma das ligações JMS constrói e envia uma mensagem para uma fila JMS e observa uma fila para a chegada da resposta, se houver.

- JMS

A ligação JMS permite acessar o provedor JMS integrado pelo WebSphere.

- JMS Genérico

A ligação JMS genérica permite acessar um sistema de mensagens do fornecedor não IBM.

- MQ JMS

A ligação JMS do MQ permite acessar o subconjunto JMS de um sistema de mensagens WebSphere MQ. Você usaria esta ligação quando o subconjunto JMS de funções é suficiente para seu aplicativo.

- MQ

A ligação do WebSphere MQ permite que você se comunique com aplicativos nativos do MQ, trazendo-os para a estrutura de arquitetura orientada a serviços e fornecendo acesso às informações de cabeçalho específicas do MQ. Você usaria esta ligação quando precisasse usar funções nativas do MQ.

Visão Geral da Ligação de Exportação e Importação

Uma exportação permite tornar os serviços em um módulo de integração disponíveis para clientes externos e uma importação torna possível para seus componentes SCA em um módulo de integração chamar serviços externos. A ligação associada à exportação ou importação especifica o relacionamento entre as mensagens de protocolo e os objetos de negócios. Ela também especifica a forma com que as operações e falhas são selecionadas.

Fluxo de Informações por meio de uma Exportação

Uma exportação recebe uma solicitação, que é destinada para o componente ao qual a exportação está ligada, por meio de um transporte específico determinado pela ligação associada (por exemplo, HTTP).

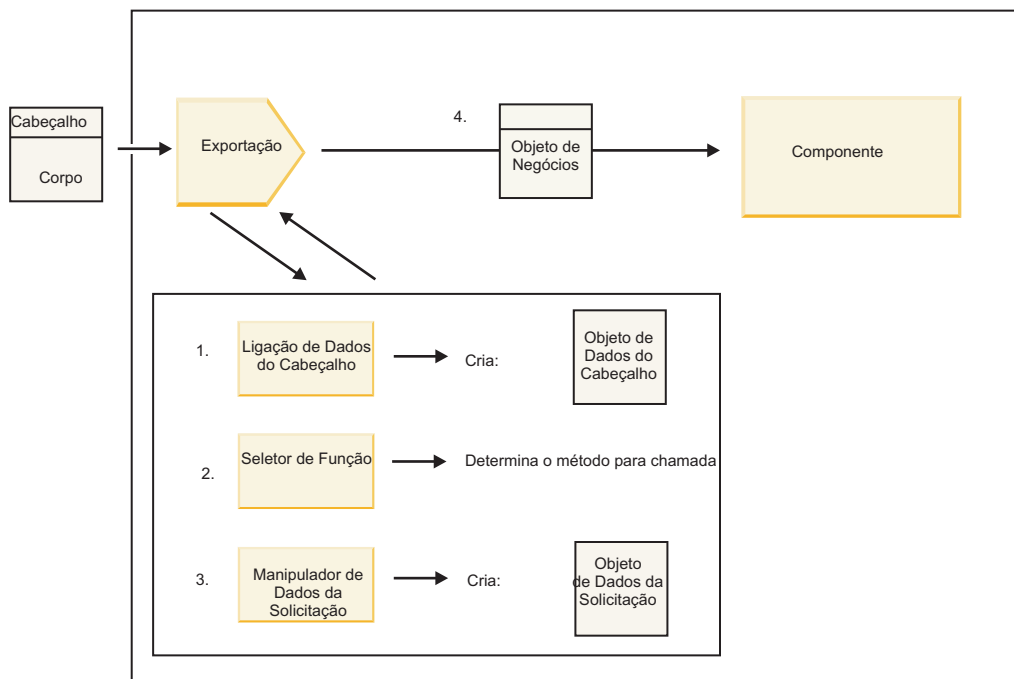


Figura 48. Fluxo de uma Solicitação por meio da Exportação para um Componente

Quando a exportação recebe a solicitação, a seguinte sequência de eventos ocorre:

1. Para ligações do WebSphere MQ apenas, a ligação de dados do cabeçalho transforma o cabeçalho do protocolo em um objeto de dados do cabeçalho.
2. O seletor de função determina o nome do método nativo da mensagem de protocolo. O nome do método nativo é mapeado pela configuração de exportação para o nome de uma operação na interface da exportação.
3. O manipulador de dados da solicitação ou a ligação de dados no método transforma a solicitação em um objeto de negócios de solicitação.
4. A exportação chama o método do componente com o objeto de negócios de solicitação.
 - A ligação de exportação HTTP, a ligação de exportação de serviço da web e a ligação de exportação EJB chamam o componente SCA de forma síncrona.
 - As ligações de exportação JMS, JMS Genéricas, JMS do MQ e do WebSphere MQ chamam o componente SCA de forma assíncrona.

Observe que uma exportação pode propagar os cabeçalhos e as propriedades do usuário que ela recebe por meio do protocolo, se a propagação do contexto está ativada. Componentes que estão ligados à

exportação podem, então, acessar esses cabeçalhos e propriedades do usuário. Consulte o tópico “Propagação” no centro de informações do WebSphere Integration Developer para obter informações adicionais.

Se esta for uma operação bidirecional, o componente retornará uma resposta.

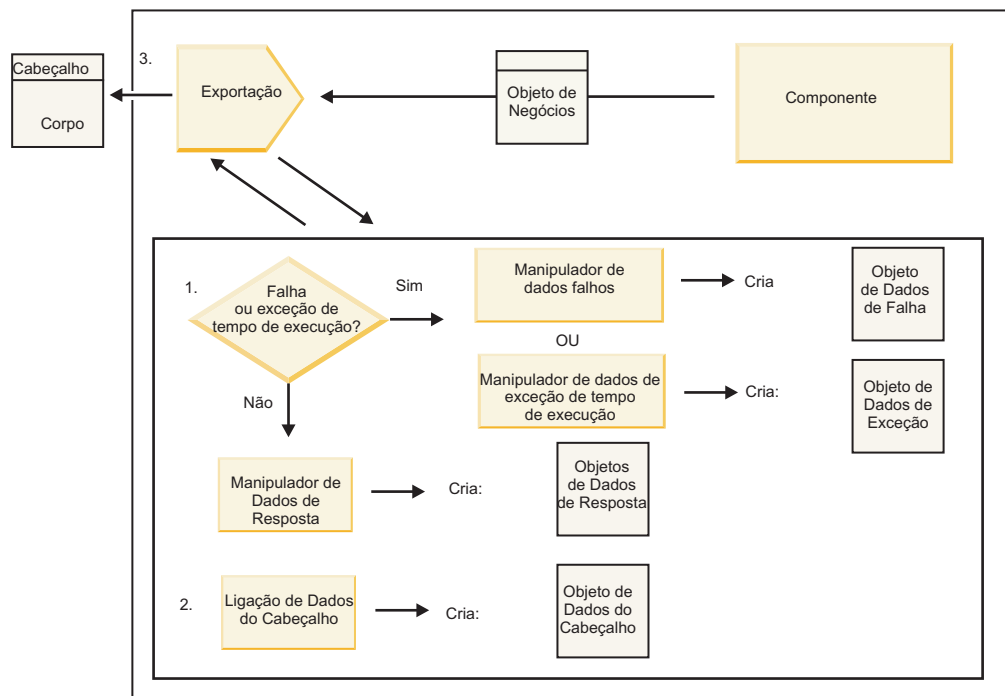


Figura 49. Fluxo de uma Resposta de Volta por meio da Exportação

A seguinte sequência de etapas ocorre:

1. Se uma mensagem de resposta normal é recebida pela ligação de exportação, o manipulador de dados de resposta ou a ligação de dados no método transforma o objeto de negócios em uma resposta. Se a resposta for uma falha, o manipulador de dados falhos ou a ligação de dados no método transformará a falha em uma resposta de falha. Para ligações de exportação HTTP apenas, se a resposta for uma exceção de tempo de execução, o manipulador de dados de exceção de tempo de execução, se configurado, será chamado.
2. Para ligações WebSphere MQ apenas, a ligação de dados de cabeçalho transforma os objeto de dados de cabeçalho em cabeçalhos de protocolo.
3. A exportação envia a resposta de serviço durante o transporte.

Fluxo de Informações por meio de uma Importação

Componentes enviam solicitações para serviços fora do módulo usando uma importação. A solicitação é enviada, por meio de um transporte específico determinado pela ligação associada.

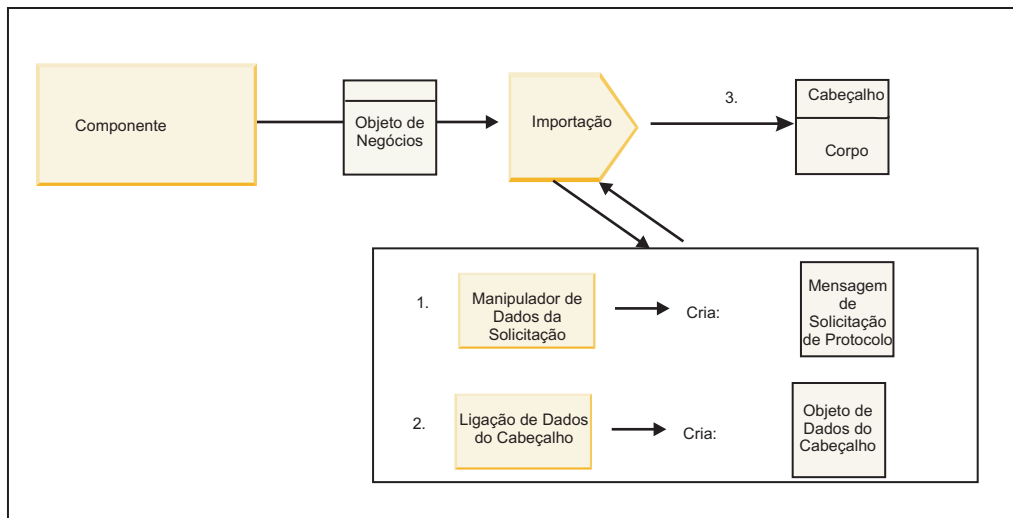


Figura 50. Fluxo a Partir de um Componente por meio da Importação para um Serviço

O componente chama a importação com um objeto de negócios de solicitação.

Nota:

- A ligação de importação HTTP, a ligação de importação do serviço da web e a ligação de importação EJB devem ser chamadas de forma síncrona chamando o componente.
- A ligação de importação JMS, JMS Genérica, JMS do MQ e do WebSphere MQ deve ser chamada assincronicamente.

Após o componente chamar a importação, a sequência de eventos a seguir ocorre:

1. O manipulador de dados da solicitação ou a ligação de dados no método transforma o objeto de negócios de solicitação em uma mensagem de solicitação de protocolo.
2. Para ligações do WebSphere MQ apenas, a ligação de dados do cabeçalho no método configura o objeto de negócios do cabeçalho no cabeçalho do protocolo.
3. A importação chama o serviço com a solicitação de serviço durante o transporte.

Se esta for uma operação bidirecional, o serviço retornará uma resposta e a sequência de etapas a seguir ocorrerá:

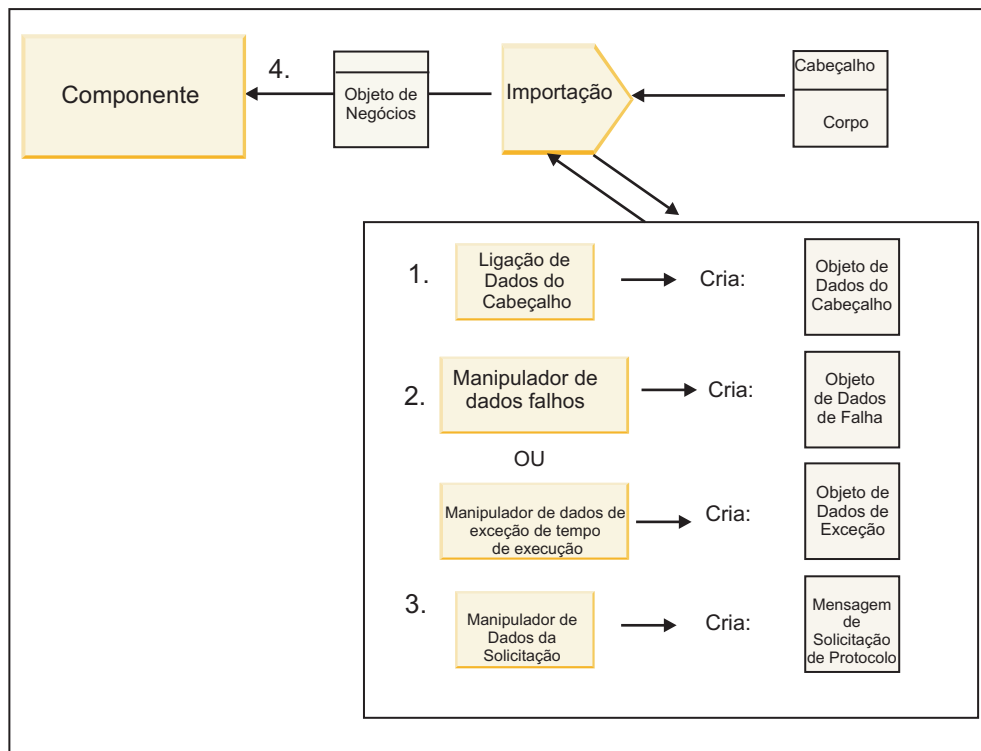


Figura 51. Fluxo de uma Resposta de Volta por meio da Importação

1. Para ligações do WebSphere MQ apenas, a ligação de dados do cabeçalho transforma o cabeçalho do protocolo em um objeto de dados do cabeçalho.
2. Uma determinação é feita sobre se a resposta é uma falha.
 - Se a resposta for uma falha, o seletor de falha inspecionará a falha para determinar para qual falha de WSDL ele mapeia. O manipulador de dados falhos no método, então, transforma a falha em uma resposta de falha.
 - Se a resposta for uma exceção de tempo de execução, o manipulador de dados de exceção de tempo de execução, se configurado, será chamado.
3. O manipulador de dados de resposta ou a ligação no método transforma a resposta em um objeto de negócios de resposta.
4. A importação retorna o objeto de negócios de resposta para o componente.

Configuração da Exportação e Importação de Ligação

Um dos aspectos principais das ligações de exportação e importação é a transformação do formato de dados, que indica como os dados são mapeados (desserializados) a partir de um formato de ligação nativo para um objeto de negócios ou como eles são mapeados (serializados) de um objeto de negócios para um formato de ligação nativo. Para ligações associadas às exportações, também é possível especificar um seletor de função para indicar qual operação deve ser executada nos dados. Para ligações associadas às exportações ou importações, é possível indicar como falhas que ocorrem durante o processamento devem ser manipuladas.

Além disso, especifique informações específicas de transporte nas ligações. Por exemplo, para uma ligação HTTP, especifique a URL do terminal. Para a ligação HTTP, as informações específicas de transporte são descritas nos tópicos “Gerando uma Ligação de Importação HTTP” e “Gerando uma Ligação de Exportação HTTP”. Também é possível localizar informações sobre outras ligações no centro de informações.

Transformação do Formato de Dados em Importações e Exportações

Quando uma ligação de exportação ou importação é configurada no IBM Integration Designer, uma das propriedades de configuração que especificar é o formato de dados usado pela ligação.

- Para ligações de exportação, nas quais um aplicativo cliente envia solicitações e recebe respostas a partir de um componente SCA, você indica o formato dos dados nativos. Dependendo do formato, o sistema seleciona o manipulador de dados ou a ligação de dados apropriada para transformar os dados nativos em um objeto de negócios (que é usado pelo componente SCA) e, inversamente, para transformar o objeto de negócios em dados nativos (que é a resposta para o aplicativo cliente).
- Para ligações de importação, nas quais um componente SCA envia solicitações e recebe respostas de um serviço fora do módulo, você indica o formato de dados dos dados nativos. Dependendo do formato, o sistema seleciona o manipulador de dados ou a ligação de dados apropriada para transformar o objeto de negócios em dado nativo e vice-versa.

O IBM Business Process Manager fornece um conjunto de formatos de dados predefinidos e manipuladores de dados ou ligações de dados correspondentes que suportam os formatos. Também é possível criar seus próprios manipuladores de dados customizados e registrar o formato de dados para esses manipuladores de dados. Para obter informações adicionais, consulte o tópico “Desenvolvendo manipuladores de dados” no centro de informações do IBM Integration Designer.

- *Manipuladores de Dados* são de protocolo neutro e transformam dados de um formato para outro. No IBM Business Process Manager, os manipuladores de dados geralmente transformam dados nativos (tais como XML, CSV e COBOL) em um objeto de negócios e um objeto de negócios em dado nativo. Como eles são de protocolo neutro, é possível reutilizar o mesmo manipulador de dados com uma variedade de ligações de exportação e importação. Por exemplo, é possível usar o mesmo manipulador de dados XML com uma ligação de exportação ou importação HTTP ou com uma ligação de exportação ou importação JMS.
- *Ligações de Dados* também transformam dados nativos em um objeto de negócios (e vice-versa), mas elas são específicas de protocolo. Por exemplo, uma ligação de dados HTTP pode ser usada apenas com uma ligação de exportação ou de importação HTTP. Diferente de manipuladores de dados, uma ligação de dados HTTP não pode ser reutilizada com uma ligação de exportação ou importação do MQ.

Nota: Três ligações de dados HTTP (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML e HTTPServiceGatewayDataBinding) estão descontinuadas a partir do IBM Business Process Manager Versão 7.0. Use manipuladores de dados sempre que possível.

Conforme observado anteriormente, é possível criar manipuladores de dados customizados, se necessário. Também é possível criar ligações de dados customizadas; no entanto, é recomendado que você crie manipuladores de dados customizados porque eles podem ser usados em diversas ligações.

Manipuladores de Dados:

Os manipuladores de dados são configurados com relação às ligações de exportação e de importação para transformar dados de um formato para outro no estilo protocolo neutro. Vários manipuladores de dados são fornecidos como parte do produto, mas também é possível criar seu próprio manipulador de dados, se necessário. É possível associar um manipulador de dados a uma ligação de exportação ou de importação em um de dois níveis: é possível associá-lo à todas as operações na interface de exportação ou de importação ou é possível associá-lo a uma operação específica para solicitação ou resposta.

Manipuladores de Dados Predefinidos

Você usa o IBM Integration Designer para especificar o manipulador de dados que deseja usar.

Os manipuladores de dados que são predefinidos para seu uso são listados na seguinte tabela, que também descreve como cada manipulador de dados transforma os dados de entrada e de saída.

Nota: Exceto onde observado, esse manipuladores de dados podem ser usados com ligações JMS, JMS Genérico, JMS do MQ, WebSphere MQ e HTTP. Consulte o tópico “Manipuladores de Dados” no centro de informações do Integration Designer para obter informações mais detalhadas.

Tabela 52. Manipuladores de Dados Predefinidos

Manipulador de dados	Dados nativos para objeto de negócios	Objeto de negócios para dados nativos
ATOM	Analisa os feeds ATOM em um objeto de negócios feed ATOM.	Serializa um objeto de negócios feed ATOM para os feeds ATOM.
Delimitado	Analisa dados delimitados em um objeto de negócios.	Serializa um objeto de negócios para dados delimitados, incluindo o CSV.
Largura Fixa	Analisa dados de largura fixa em um objeto de negócios.	Serializa um objeto de negócios para dados de largura fixa.
Manipulado pelo WTX	Delega transformação de formato de dados para o WebSphere Transformation Extender (WTX). O nome do mapa WTX é derivado pelo manipulador de dados.	Delega transformação de formato de dados para o WebSphere Transformation Extender (WTX). O nome do mapa WTX é derivado pelo manipulador de dados.
Manipulado pelo invocador de WTX	Delega transformação de formato de dados para o WebSphere Transformation Extender (WTX). O nome do mapa WTX é fornecido pelo usuário.	Delega transformação de formato de dados para o WebSphere Transformation Extender (WTX). O nome do mapa WTX é fornecido pelo usuário.
JAXB	Serializa os Java beans para um objeto de negócios usando as regras de mapeamento definidas pela especificação Java Architecture for XML Binding (JAXB).	Desserializa um objeto de negócios para os Java beans usando as regras de mapeamento definidas pela especificação JAXB.
JAXWS Nota: O manipulador de dados do JAXWS pode ser usado apenas com a ligação EJB.	Usado por uma ligação EJB para transformar um objeto Java de resposta ou um objeto Java de exceção para um objeto de negócios de resposta usando as regras de mapeamento definidas pela especificação Java API for XML Web Services (JAX-WS).	Usado por uma ligação EJB para transformar um objeto de negócios para os parâmetros do método Java de saída usando as regras de mapeamento definidas pela especificação JAX-WS.
JSON	Analisa dados JSON em um objeto de negócios.	Serializa um objeto de negócios para dados JSON.
Corpo nativo	Analisa os bytes, texto, mapa, fluxo ou objetos nativos em um dos cinco objetos de negócios de base (texto, bytes, mapa, fluxo ou objeto).	Transforma os cinco objetos de negócios de base em byte, texto, mapa, fluxo ou objeto.
SOAP	Analisa a mensagem SOAP (e o cabeçalho) em um objeto de negócios.	Serializa um objeto de negócios para uma mensagem SOAP.
XML	Analisa dados XML em um objeto de negócios.	Serializa um objeto de negócios para dados XML.
UTF8XMLDataHandler	Analisa dados XML codificados em UTF-8 em um objeto de negócios.	Serializa um objeto de negócios em dados XML codificados em UTF-8 quando enviar uma mensagem.

Criando um Manipulador de Dados

Informações detalhadas sobre a criação de um manipulador de dados podem ser localizadas no tópico “Desenvolvendo Manipuladores de Dados” no centro de informações do Integration Designer.

Ligações de Dados:

As ligações de dados são configuradas nas ligações de exportação e importação para transformar dados de um formato para outro. As ligações de dados são específicas de um protocolo. Várias ligações de dados são fornecidas como parte do produto, mas também é possível criar sua própria ligação de dados, se necessário. É possível associar uma ligação de dados a uma ligação de exportação ou importação em um de dois níveis - é possível associá-la a todas as operações na interface da exportação ou importação ou você pode associá-la a uma operação específica para a solicitação ou resposta.

É possível usar o IBM Integration Designer para especificar qual ligação de dados deseja usar ou criar sua própria ligação de dados. Uma discussão de como criar ligações de dados pode ser localizada na seção “Visão Geral das Ligações de JMS, MQ JMS e JMS Genérica” do centro de informações do IBM Integration Designer.

Ligações JMS

A tabela a seguir lista as ligações de dados que podem ser usadas com:

- Ligações JMS
- Ligações JMS Genéricas
- Ligações JMS do WebSphere MQ

A tabela também inclui uma descrição das tarefas que as ligações de dados executam.

Tabela 53. Ligações de Dados Predefinidas para Ligações JMS

Ligação de Dados	Dados nativos para objeto de negócios	Objeto de negócios para dados nativos
Objeto Java Serializado	Transforma o objeto serializado Java em um objeto de negócios (que é mapeado como o tipo de entrada ou saída no WSDL).	Serializa um objeto de negócios para o objeto serializado Java na mensagem do objeto JMS.
Bytes agrupados	Extrai os bytes da mensagem de bytes JMS recebida e os agrupa no objeto de negócios JMSBytesBody.	Extrai os bytes do objeto de negócios JMSBytesBody e os agrupa na mensagem de bytes JMS de saída.
Entrada de mapa agrupada	Extrai o nome, valor e as informações de tipo para cada entrada na mensagem do mapa JMS recebida e cria uma lista de objetos de negócios MapEntry. Em seguida, agrupa a lista no objeto de negócios JMSMapBody.	Extrai o nome, o valor e as informações de tipo da lista MapEntry no objeto de negócios JMSMapBody e cria as entradas correspondentes na mensagem do mapa JMS de saída.
Objeto agrupado	Extrai o objeto da mensagem do objeto JMS recebida e o agrupa no objeto de negócios JMSObjectBody.	Extrai o objeto do objeto de negócios JMSObjectBody e o agrupa na mensagem do objeto JMS de saída.
Texto agrupado	Extrai o texto da mensagem de texto de JMS recebida e o agrupa no objeto de negócios JMSTextBody.	Extrai o texto do objeto de negócios JMSTextBody e o agrupa na mensagem de texto de JMS de saída.

Ligações WebSphere MQ

A tabela a seguir lista as ligações de dados que podem ser usadas com o WebSphere MQ e descreve as tarefas que as ligações de dados executam.

Tabela 54. Ligações de Dados Predefinidas para Ligações do WebSphere MQ

Ligação de Dados	Dados nativos para objeto de negócios	Objeto de negócios para dados nativos
Objeto Java Serializado	Transforma o objeto serializado Java a partir da mensagem recebida em um objeto de negócios (que é mapeado como o tipo de entrada ou saída no WSDL).	Transforma um objeto de negócios no objeto serializado Java na mensagem de saída
Bytes agrupados	Extrai os bytes da mensagem de bytes do MQ não estruturada e os agrupa no objeto de negócios JMSBytesBody.	Extrai os bytes de um objeto de negócios JMSBytesBody e agrupa os bytes na mensagem de bytes do MQ não estruturada de saída.
Texto agrupado	Extrai o texto de uma mensagem de texto do MQ não estruturada e o agrupa em um objeto de negócios JMSTextBody.	Extrai texto a partir de um objeto de negócios JMSTextBody e o agrupa em uma mensagem de texto do MQ não estruturada.
Entrada do fluxo agrupada	Extrai as informações de nome e tipo para toda entrada na mensagem de fluxo do JMS recebida e cria uma lista dos objetos de negócios StreamEntry. Em seguida, agrupa a lista no objeto de negócios JMSStreamBody.	Extrai as informações de nome e tipo da lista StreamEntry no objeto de negócios JMSStreamBody e cria entradas correspondentes na JMSStreamMessage de saída.

Além das ligações de dados listadas em Tabela 25 na página 62, o WebSphere MQ também usa ligações de dados de cabeçalho. Consulte o centro de informações do IBM Integration Designer para obter detalhes.

Ligações HTTP

A tabela a seguir lista as ligações de dados que podem ser usadas com HTTP e descreve as tarefas executadas pelas ligações de dados.

Tabela 55. Ligações de dados predefinidas para ligações HTTP

Ligação de Dados	Dados nativos para objeto de negócios	Objeto de negócios para dados nativos
Bytes agrupados	Extrai os bytes do corpo da mensagem HTTP recebida e os agrupa no objeto de negócios HTTPBytes.	Extrai os bytes do objeto de negócios HTTPBytes e os inclui no corpo da mensagem HTTP de saída.
Texto agrupado	Extrai o texto do corpo da mensagem HTTP recebida e o agrupa no objeto de negócios HTTPText.	Extrai o texto do objeto de negócios HTTPText e o inclui no corpo da mensagem HTTP de saída.

Seletores de Função em Ligações de Exportação

Um seletor de função é usado para indicar qual operação deve ser executada nos dados para uma mensagem de solicitação. Os seletores de função são configurados como parte de uma ligação de exportação.

Considere uma exportação SCA que expõe uma interface. A interface contém duas operações - Criar e Atualizar. A exportação tem uma ligação JMS que lê a partir de uma fila.

Quando uma mensagem chega na fila, a exportação transmite os dados associados, mas qual operação da interface de exportação deve ser chamada no componente ligado? A operação é determinada pelo seletor de função e pela configuração de ligação de exportação.

O seletor de função retorna o nome da função nativa (o nome da função no sistema do cliente que enviou a mensagem). O nome da função nativa é, então, mapeado para o nome da operação ou função na interface associada à exportação. Por exemplo, na figura a seguir, o seletor de função retorna o nome da função nativa (CRT) a partir da mensagem recebida, o nome da função nativa é mapeado para a operação Criar e o objeto de negócios é enviado para o componente SCA com a operação Criar.

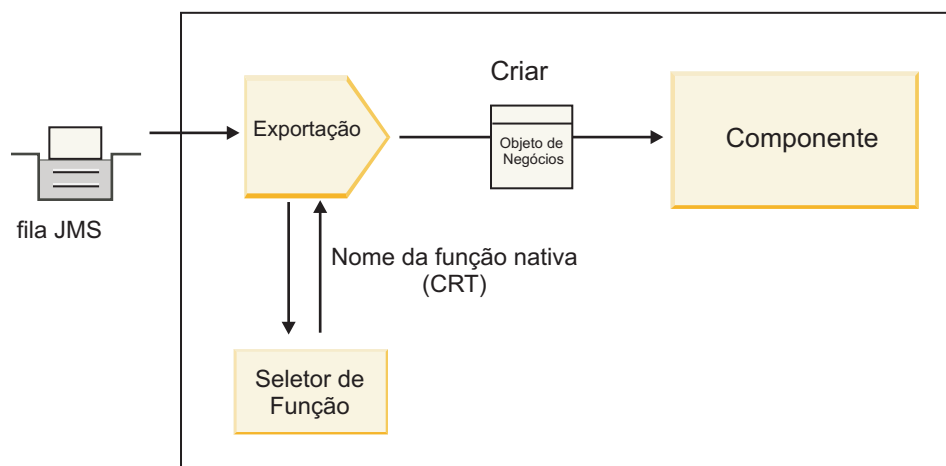


Figura 52. O Seletor de Função

Se a interface tiver apenas uma operação, não há necessidade de especificar um seletor de função.

Vários seletores de função pré-empacotados estão disponíveis e são listados nas seções a seguir.

Ligações JMS

A tabela a seguir lista os seletores de função que podem ser usados com:

- Ligações JMS
- Ligações JMS Genéricas
- Ligações JMS do WebSphere MQ

Tabela 56. Seletores de Função Predefinidos para Ligações JMS

Seletor de Função	Descrição
Seletor de Função JMS para Ligações de Dados do JMS Simples	Usa a propriedade JMSType da mensagem para selecionar o nome da operação.
Seletor de Função de Propriedade do Cabeçalho JMS	Retorna o valor da Propriedade de Sequência JMS, TargetFunctionName, do cabeçalho.
Seletor de Função do Gateway de Serviço JMS	Determina se a solicitação é uma operação unidirecional ou bidirecional examinando a propriedade JMSReplyTo configurada pelo cliente.

Ligações do WebSphere MQ

A tabela a seguir lista os seletores de função que podem ser usados com ligações do WebSphere MQ.

Tabela 57. Seletores de Função Predefinidos para Ligações do WebSphere MQ

Seletor de Função	Descrição
Seletor de Função handleMessage MQ	Retorna handleMessage como um valor, que é mapeado usando as ligações de método de exportação para o nome de uma operação na interface.
O MQ utiliza o seletor da função padrão JMS	Lê a operação nativa a partir da propriedade TargetFunctionName da pasta de um cabeçalho MQRFH2.
O MQ utiliza o formato do corpo da mensagem como função nativa	Localiza o campo Formato do último cabeçalho e retorna esse campo como uma Sequência.
Seletor de Função de Tipo MQ	Cria um método em sua ligação de exportação recuperando uma URL contendo as propriedades Msd, Set, Type e Format localizadas no cabeçalho MQRFH2.
Seletor de Função do Gateway de Serviço MQ	Usa a propriedade MsgType no cabeçalho MQMD para determinar o nome da operação.

Ligações HTTP

A tabela a seguir lista os seletores de função que podem ser usados com ligações HTTP.

Tabela 58. Seletores de Função Predefinidos para Ligações HTTP

Seletor de Função	Descrição
Seletor de Função HTTP com base em Cabeçalho TargetFunctionName	Usa a propriedade do cabeçalho de HTTP TargetFunctionName do cliente para determinar qual operação chamar no tempo de execução da exportação.
Seletor de Função HTTP Baseado na URL e no Método de HTTP	Usa o caminho relativo da URL anexada com o método de HTTP a partir do cliente para determinar a operação nativa definida na exportação.
Seletor de função de gateway de serviço HTTP baseado na URL com um nome da operação	Determina o método a ser chamado com base na URL se "operationMode = oneway" tiver sido anexado à URL de solicitação.

Nota: Também é possível criar seu próprio seletor de função, usando o IBM Integration Designer. Informações sobre como criar um seletor de função são fornecidas no centro de informações do IBM Integration Designer. Por exemplo, uma descrição de como criar um seletor de função para ligações do WebSphere MQ pode ser localizada em "Visão Geral dos Seletores de Função do MQ".

Manipulação de Falha

É possível configurar suas ligações de importação e exportação para manipular falhas (por exemplo, exceções de negócios) que ocorrem durante o processamento especificando manipuladores de dados falhos. É possível configurar um manipulador de dados falhos em três níveis - você pode associar um manipulador de dados falhos a uma falha, a uma operação ou para todas as operações a uma ligação.

Um manipulador de dados falhos processa dados falhos e os transforma no formato correto para serem enviados pela ligação de exportação ou importação.

- Para uma ligação de exportação, o manipulador de dados falhos transforma o objeto de negócios de exceção enviado do componente para uma mensagem de resposta que pode ser usada pelo aplicativo cliente.

- Para uma ligação de importação, o manipulador de dados falhos transforma os dados falhos ou a mensagem de resposta enviada de um serviço para um objeto de negócios de exceção que pode ser usado pelo componente SCA.

Para ligações de importação, a ligação chama o seletor de falha, que determina se a mensagem de resposta é uma resposta normal, uma falha de negócios ou uma exceção de tempo de execução.

É possível especificar um manipulador de dados falhos para uma determinada falha, para uma operação e para todas as operações com uma ligação.

- Se o manipulador de dados falhos estiver configurado em todos os três níveis, o manipulador de dados associado a uma determinada falha será chamado.
- Se os manipuladores de dados falhos estiverem configurados nos níveis da operação e da ligação, o manipulador de dados associado à operação será chamado.

Dois editores são usados no IBM Integration Designer para especificar a manipulação de falhas. O editor de interface é usado para indicar se haverá uma falha em uma operação. Depois que uma ligação é gerada com esta interface, o editor na visualização de propriedades permite configurar como o falha será manipulada. Para obter informações adicionais, consulte o tópico “Seletores de Falha” no centro de informações do IBM Integration Designer.

Como as Falhas São Manipuladas em Ligações de Exportação:

Quando ocorre uma falha durante o processamento da solicitação de um aplicativo cliente, a ligação de exportação pode retornar as informações de falha para o cliente. Configure a ligação de exportação para especificar como a falha deve ser processada e retornada ao cliente.

Configure a ligação de exportação usando IBM Integration Designer.

Durante o processamento da solicitação, um cliente chama uma exportação com uma solicitação e a exportação chama o componente SCA. Durante o processamento da solicitação, o componente SCA pode retornar uma resposta de negócios ou pode emitir uma exceção de negócios de serviço ou uma exceção de tempo de execução de serviço. Quando isso ocorre, a ligação de exportação transforma a exceção em uma mensagem de falha e a envia para o cliente, conforme mostrado na figura a seguir e descrito nas seções a seguir.

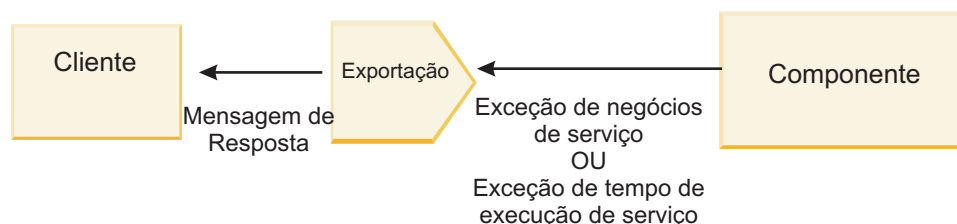


Figura 53. Como Informações de Falha são Enviadas a Partir do Componente Através da Ligação de Exportação para o Cliente

É possível criar um manipulador de dados ou uma ligação de dados customizada para manipular falhas.

Falhas de Negócios

Falhas de negócios são erros de negócios ou exceções que ocorrem durante o processamento.

Considere a interface a seguir, que possui uma operação createCustomer nela. Esta operação possui duas falhas de negócios definidas: CustomerAlreadyExists e MissingCustomerId.

Operações

Operações e seus parâmetros

	Nome	Tipo
createCustomer		
Entrada(s)	entrada	CustomerInfo
Saída(s)	saída	CustomerInfo
Falha	Cliente Já Existe	Customer Already ExistsBO
Falha	MissingCustomerID	MissingCustomerIDBO

Figura 54. Interface com Duas Falhas

Neste exemplo, se um cliente enviar uma solicitação para criar um cliente (para este componente SCA) e se o cliente já existir, o componente lançará uma falha `CustomerAlreadyExists` para a exportação. A exportação precisa propagar esta falha de negócios de volta para o cliente de chamada. Para fazer isso, ela usa o manipulador de dados falhos que está configurado na ligação de exportação.

Quando uma falha de negócios é recebida pela ligação de exportação, ocorre o seguinte processamento:

1. A ligação determina qual manipulador de dados falhos chamar para manipular a falha. Se a exceção de negócios de serviço contém o nome da falha, o manipulador de dados que está configurado na falha é chamado. Se a exceção de negócios de serviço não contém o nome da falha, o nome da falha é derivado correspondendo os tipos de falha.
2. A ligação chama o manipulador de dados falhos com o objeto de dados da exceção de negócios de serviço.
3. O manipulador de dados falhos transforma o objeto de dados falhos em uma mensagem de resposta e a retorna para a ligação de exportação.
4. A exportação retorna a mensagem de resposta para o cliente.

Se a exceção de negócios de serviço contém o nome da falha, o manipulador de dados que está configurado na falha é chamado. Se a exceção de negócios de serviço não contém o nome da falha, o nome da falha é derivado correspondendo os tipos de falha.

Exceções de Tempo de Execução

Uma exceção de tempo de execução é uma exceção que ocorre no aplicativo SCA durante o processamento de uma solicitação que não corresponde a uma falha de negócios. Ao contrário de falhas de negócios, exceções de tempo de execução não são definidas na interface.

Em determinados cenários, você pode desejar propagar essas exceções de tempo de execução para o aplicativo cliente para que o aplicativo cliente possa executar a ação apropriada.

Por exemplo, se um cliente envia uma solicitação (para o componente SCA) para criar um cliente e um erro de autorização ocorre durante o processamento da solicitação, o componente emite uma exceção de tempo de execução. Esta exceção de tempo de execução precisa ser propagada de volta ao cliente de chamada para que ele possa executar a ação apropriada com relação à autorização. Isso é realizado pelo manipulador de dados de exceção de tempo de execução configurado na ligação de exportação.

Nota: É possível configurar um manipulador de dados de exceção de tempo de execução somente em ligações HTTP.

O processamento de uma exceção de tempo de execução é semelhante ao processamento de uma falha de negócios. Se um manipulador de dados de exceção de tempo de execução foi configurado, o processamento a seguir ocorrerá:

1. A ligação de exportação chama o manipulador de dados apropriado com a exceção de tempo de execução de serviço.
2. O manipulador de dados transforma o objeto de dados falhos em uma mensagem de resposta e a retorna para a ligação de exportação.
3. A exportação retorna a mensagem de resposta para o cliente.

A manipulação de falha e a manipulação de exceção de tempo de execução são opcionais. Se você não desejar propagar falhas ou exceções de tempo de execução para o cliente de chamada, não configure o manipulador de dados falhos ou o manipulador de dados de exceção de tempo de execução.

Como as Falhas São Manipuladas em Ligações de Importação:

Um componente usa uma importação para enviar uma solicitação para um serviço fora do módulo. Quando ocorre uma falha durante o processamento da solicitação, o serviço retorna a falha para a ligação de importação. É possível configurar a ligação de importação para especificar como a falha deve ser processada e retornada ao componente.

Configure a ligação de importação usando o IBM Integration Designer. É possível especificar um manipulador de dados falhos (ou ligação de dados) e você também especifica um seletor de falha.

Manipuladores de Dados Falhos

O serviço que processa a solicitação envia, para a ligação de importação, informações de falha na forma de uma exceção ou uma mensagem de resposta que contém os dados da falha.

A ligação de importação transforma a exceção de serviço ou mensagem de resposta em uma exceção de negócios do serviço ou exceção de tempo de execução de serviço, conforme mostrado na figura a seguir e descrito nas seções a seguir.

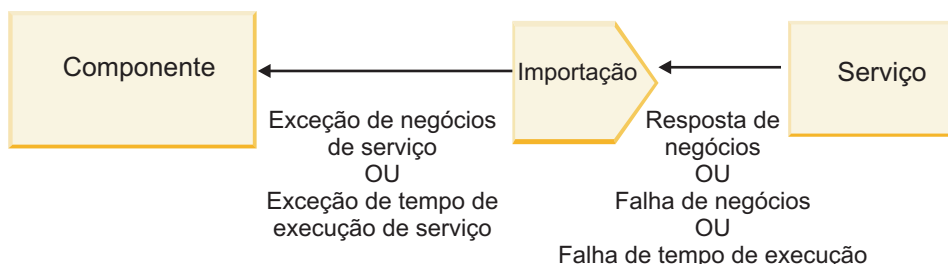


Figura 55. Como as Informações de Falha são Enviadas a Partir do Serviço Através da Importação para o Componente

É possível criar um manipulador de dados ou uma ligação de dados customizada para manipular falhas.

Seletores de Falha

Quando você configura uma ligação de importação, pode especificar um seletor de falha. O seletor de falha determina se a resposta de importação é uma resposta real, uma exceção de negócios ou uma falha de tempo de execução. Ele também determina, a partir do corpo de resposta ou cabeçalho, o nome da falha nativa, que é mapeado pela configuração de ligação para o nome de uma falha na interface associada.

Dois tipos de seletores de falha pré-empacotados estão disponíveis para uso com as importações de JMS, JMS do MQ, JMS Genéricas, do WebSphere MQ e HTTP:

Tabela 59. Seletores de Falha Pré-empacotados

Tipo de Seletor de Falha	Descrição
Baseado em Cabeçalho	Determina se uma mensagem de resposta é uma falha de negócios, uma exceção de tempo de execução ou uma mensagem normal baseada nos cabeçalhos na mensagem de resposta recebida.
SOAP	Determina se a mensagem SOAP de resposta é uma resposta normal, falha de negócios ou exceção de tempo de execução.

A seguir são mostrados exemplos de seletores de falha baseados em cabeçalho e do seletor de falha de SOAP.

- Seletor de Falha Baseado em Cabeçalho

Se um aplicativo deseja indicar que a mensagem de entrada é uma falha de negócios, deve haver dois cabeçalhos na mensagem de entrada para falhas de negócios, o que é mostrado como segue:

```
Header name = FaultType, Header value = Business
Header name = FaultName, Header value = <user defined native fault name>
```

Se um aplicativo desejar indicar que a mensagem de resposta recebida é uma exceção de tempo de execução, deverá haver um cabeçalho na mensagem recebida, o que é mostrado como segue:

```
Header name = FaultType, Header value = Runtime
```

- Seletor de falhas de SOAP

Uma falha de negócios pode ser enviada como parte da mensagem SOAP com o seguinte cabeçalho SOAP customizado. "CustomerAlreadyExists" é o nome da falha neste caso.

```
<ibmSoap:BusinessFaultName
xmlns:ibmSoap="http://www.ibm.com/soap">CustomerAlreadyExists
</ibmSoap:BusinessFaultName>
```

O seletor de falha é opcional. Se você não especificar um seletor de falha, a ligação de importação não poderá determinar o tipo de resposta. A ligação, portanto, a tratará como uma resposta de negócios e chamará o manipulador de dados de resposta ou ligação de dados.

É possível criar um seletor de falha customizado. As etapas para criar um seletor de falha customizado são fornecidas no tópico "Desenvolvendo um Seletor de Falha Customizado" do centro de informações do IBM Integration Designer.

Falhas de Negócios

Uma falha de negócios pode ocorrer quando há um erro no processamento de uma solicitação. Por exemplo, se você enviar uma solicitação para criar um cliente e esse cliente já existir, o serviço enviará uma exceção de negócios para a ligação de importação.

Quando uma exceção de negócios é recebida pela ligação, as etapas de processamento dependem de se um seletor de falha foi configurado para a ligação.

- Se nenhum seletor de falha foi configurado, a ligação chamará a ligação de dados ou manipulador de dados de resposta.
- Se um seletor de falha foi configurado, ocorrerá o seguinte processamento:
 1. A ligação de importação chama o seletor de falha para determinar se a resposta é falha de negócios, resposta de negócios ou falha no tempo de execução.
 2. Se a resposta for uma falha de negócios, a ligação de importação chamará o seletor de falha para fornecer o nome de falha nativo.
 3. A ligação de importação determina a falha WSDL correspondente ao nome de falha nativo retornado pelo seletor de falha.

4. A ligação de importação determina o manipulador de dados falhos que está configurado para esta falha de WSDL.
5. A ligação de importação chama este manipulador de dados falhos com os dados falhos.
6. O manipulador de dados falhos transforma os dados falhos em um objeto de dados e o retorna para a ligação de importação.
7. A ligação de importação constrói um objeto de exceção de negócios de serviço com o objeto de dados e o nome da falha.
8. A importação retorna o objeto de exceção de negócios de serviço para o componente.

Exceções de Tempo de Execução

Uma exceção de tempo de execução pode ocorrer quando há um problema na comunicação com o serviço. O processamento de uma exceção de tempo de execução é semelhante ao processamento de uma exceção de negócios. Se um seletor de falha foi configurado, ocorrerá o seguinte processamento:

1. A ligação de importação chama o manipulador de dados de exceção de tempo de execução apropriado com os dados da exceção.
2. O manipulador de dados de exceção de tempo de execução transforma os dados de exceção em um objeto de exceção de tempo de execução de serviço e o retorna para a ligação de importação.
3. A importação retorna o objeto de exceção de tempo de execução de serviço para o componente.

Interoperabilidade entre módulos SCA e serviços SCA abertos

O IBM WebSphere Application Server v7.0 Feature Pack para Service Component Architecture (SCA) fornece um modelo de programação simples, porém poderoso, para construir aplicativos com base nas especificações de SCA Aberto. Os módulos SCA do IBM Business Process Manager usam ligações de importação e exportação para interoperar com serviços de SCA Aberto desenvolvidos em um ambiente do Rational Application Developer e hospedados pelo WebSphere Application Server Feature Pack para Service Component Architecture.

Um aplicativo SCA chama um aplicativo SCA Aberto por meio de uma ligação de importação. Um aplicativo SCA recebe uma chamada de um aplicativo SCA Aberto por meio de uma ligação de exportação. Uma lista de ligações suportadas é mostrada no “Chamando Serviços sobre Ligações Interoperáveis” na página 71.

Chamando os Serviços de SCA Aberto a Partir de Módulos SCA

Aplicativos SCA desenvolvidos com o IBM Integration Designer podem chamar aplicativos SCA Abertos desenvolvidos em um ambiente do Rational Application Developer. Esta seção fornece um exemplo de como chamar um serviço SCA Aberto a partir de um módulo SCA usando uma ligação de importação SCA.

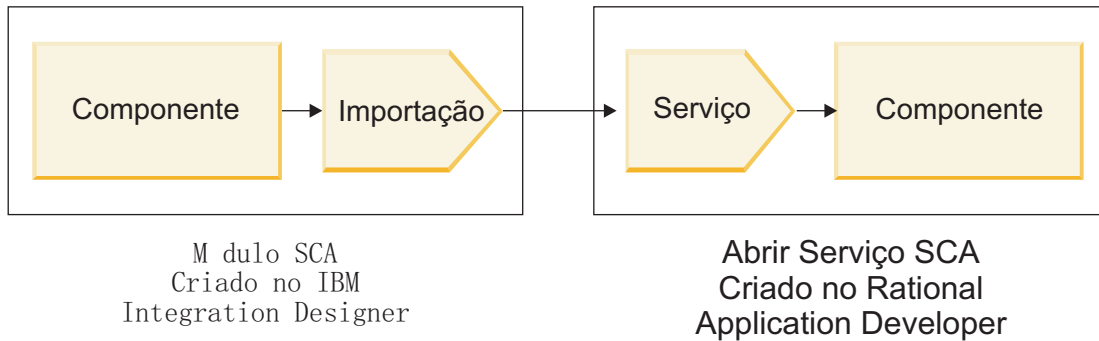


Figura 56. Componente no Módulo SCA Chamando Serviço SCA Aberto

Nenhuma configuração especial é necessária para chamar um serviço SCA Aberto.

Para se conectar a um serviço SCA Aberto por meio de uma ligação de importação SCA, você fornece o nome do componente e nome do serviço SCA Aberto na ligação de importação.

1. Para obter o nome do componente de destino e serviço do SCA Aberto composto, execute as seguintes etapas:
 - a. Assegure que a guia **Propriedades** esteja aberta clicando em **Janela > Mostrar Visualização > Propriedades**.
 - b. Abra o editor composto dando um clique duplo no diagrama composto que contém o componente e o serviço. Por exemplo, para um componente denominado **customer**, o diagrama composto é **customer.composite_diagram**.
 - c. Clique no componente de destino.
 - d. No campo **Nome** da guia **Propriedades**, anote o nome do componente de destino.
 - e. Clique no ícone de serviço associado ao componente.
 - f. No campo **Nome** da guia **Propriedades**, anote o nome do serviço.
2. Para configurar a importação do IBM Business Process Manager para conectá-lo ao serviço SCA Aberto, execute as seguintes etapas:
 - a. No IBM Integration Designer, navegue até a guia **Propriedades** da importação SCA que você deseja conectar ao serviço SCA Aberto.
 - b. No campo **Nome do Módulo**, insira o nome do componente a partir da etapa 1d na página 70.
 - c. No campo **Nome de Exportação**, insira o nome do serviço a partir da etapa 1f na página 70.
 - d. Salve seu trabalho pressionando Ctrl+S.

Chamando Módulos SCA a Partir de Serviços SCA Abertos

Aplicativos SCA Abertos desenvolvidos em um ambiente do Rational Application Developer podem chamar aplicativos SCA desenvolvidos com o IBM Integration Designer. Esta seção fornece um exemplo de como chamar um módulo SCA (por meio de uma ligação de exportação SCA) a partir de um serviço SCA Aberto.

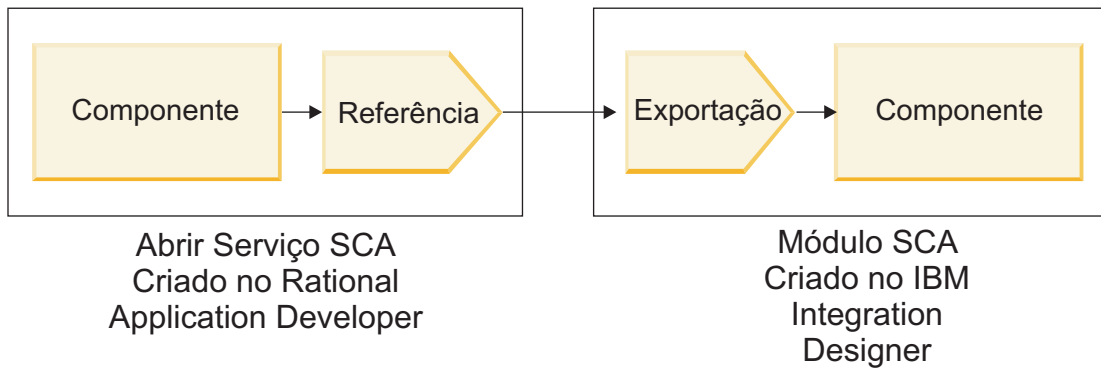


Figura 57. Serviço SCA Aberto Chamando Componente no Módulo SCA

Para se conectar a um componente SCA por meio de uma ligação de referência de SCA Aberto, você fornece o nome do módulo e o nome de exportação.

1. Para obter o nome do módulo de destino e da exportação, execute as seguintes etapas:
 - a. No IBM Integration Designer, abra o módulo no editor de montagem dando um clique duplo no módulo.
 - b. Clique na exportação.
 - c. No campo **Nome** da guia **Propriedades**, anote o nome da exportação.
2. Configure a referência de SCA Aberto que você deseja conectar ao módulo e à exportação do IBM Business Process Manager:
 - a. No Rational Application Developer, abra o editor composto dando um clique duplo no diagrama composto que contém o componente e o serviço.
 - b. Clique no ícone de referência da referência de componente para exibir as propriedades de referência na guia **Propriedades**.
 - c. Clique na guia **Ligação** no lado esquerdo da página.
 - d. Clique em **Ligações** e, em seguida, clique em **Incluir**.
 - e. Selecione a ligação **SCA**.
 - f. No campo **URI**, insira o nome do módulo do IBM Business Process Manager, seguido de uma barra ("/"), seguido pelo nome de exportação (que você determinou na etapa 1c na página 71).
 - g. Clique em **OK**.
 - h. Salve seu trabalho pressionando Ctrl+S.

Chamando Serviços sobre Ligações Interoperáveis

As seguintes ligações são suportadas para interoperabilidade com um serviço SCA Aberto.

- **Ligação SCA**

No IBM Business Process Manager, quando um módulo SCA chama um serviço SCA Aberto por meio de uma ligação de importação SCA, os seguintes estilos de chamada são suportados:

- Assíncrono (unidirecional)
- Síncrono (solicitação/resposta)

A interface de importação SCA e a interface de serviço Open SCA devem usar uma interface WSDL compatível com interoperabilidade de serviços da web (WS-I).

Observe que a ligação SCA suporta propagação de transação e contexto de segurança.

- **Ligação de serviço da Web (JAX-WS) com o protocolo SOAP1.1/HTTP ou SOAP1.2/HTTP**

A interface de importação SCA e a interface de serviço Open SCA devem usar uma interface WSDL compatível com interoperabilidade de serviços da web (WS-I).

Além disso, as seguintes qualidades de serviço são suportadas:

- Transação Atômica de Serviços da Web
- Segurança de Serviços da Web
- Ligação EJB

Uma interface Java é usada para definir a interação entre um módulo SCA e um serviço SCA Aberto quando a ligação EJB é usada.

Observe que a ligação EJB suporta propagação de transação e contexto de segurança.
- Ligações JMS

A interface de importação SCA e a interface de serviço Open SCA devem usar uma interface WSDL compatível com interoperabilidade de serviços da web (WS-I).

Os seguintes provedores JMS são suportados:

 - WebSphere Platform Messaging (Ligação JMS)
 - WebSphere MQ (Ligação JMS do MQ)

Nota: Gráficos de Negócios não são interoperáveis em nenhuma ligação SCA e, portanto, não são suportados em interfaces usadas para interoperar com o WebSphere Application Server Feature Pack para Service Component Architecture.

Tipos de Ligação

As *ligações* específicas do protocolo são usadas com importações e exportações para especificar o meio de transportar dados dentro ou fora de um módulo.

Selecionando Ligações Apropriadas

Ao criar um aplicativo, é necessário saber como selecionar a ligação mais apropriada às necessidades do seu aplicativo.

As ligações disponíveis no IBM Integration Designer fornecem uma série de opções. Nessa lista, é possível determinar qual tipo de ligação pode ser a mais adequada às necessidades do seu aplicativo.

Considere uma ligação *Service Component Architecture (SCA)* quando esses fatores são aplicáveis:

- Todos os serviços estão contidos em módulos, ou seja, não há nenhum serviço externo.
- Você deseja separar a função em diferentes módulos SCA que interagem diretamente entre si.
- Os módulos estão firmemente acoplados.

Considere uma ligação de *serviço da web* quando estes fatores são aplicáveis:

- É necessário acessar um serviço externo sobre a Internet ou fornecer um serviço sobre a Internet.
- Os serviços estão fracamente acoplados.
- É preferível a comunicação síncrona, isto é, um pedido de um serviço pode aguardar pela resposta de outro.
- O protocolo do serviço externo que você está acessando ou o serviço que você deseja fornecer é SOAP/HTTP ou SOAP/JMS.

Considere uma ligação *HTTP* quando esses fatores forem aplicáveis:

- É necessário acessar um serviço externo sobre a Internet ou fornecer um serviço sobre a Internet e você está trabalhando com outros serviços da web como GET, PUT e DELETE.
- Os serviços estão fracamente acoplados.
- É preferível a comunicação síncrona, isto é, um pedido de um serviço pode aguardar pela resposta de outro.

Considere uma ligação *Enterprise JavaBeans (EJB)* quando esses fatores são aplicáveis:

- A ligação é para um serviço importado que é por si só um EJB ou que precisa ser acessado por clientes EJB.

- O serviço importado está fracamente acoplado.
- As interações EJB stateful não são necessárias.
- É preferível a comunicação síncrona, isto é, um pedido de um serviço pode aguardar pela resposta de outro.

Considere uma ligação *Enterprise Information Systems (EIS)* quando esses fatores são aplicáveis:

- É necessário acessar um serviço em um sistema EIS usando um adaptador de recursos.
- A transmissão de dados síncrona é preferencial sobre a assíncrona.

Considere uma ligação *Java Message Service (JMS)* quando esses fatores são aplicáveis:

Importante: Existem vários tipos de ligações JMS. Se espera trocar mensagens SOAP usando JMS, considere a ligação do serviço da web com o protocolo SOAP/JMS. Consulte o “Ligações de Serviço da Web” na página 74.

- É necessário acessar um sistema de mensagens.
- Os serviços estão fracamente acoplados.
- A transmissão de dados assíncrona é preferencial sobre a síncrona.

Considere uma ligação *Generic Java Message Service (JMS)* quando esses fatores são aplicáveis:

- É necessário acessar um sistema de mensagens do fornecedor não IBM.
- Os serviços estão fracamente acoplados.
- A confiabilidade é mais importante do que o desempenho; ou seja, a transmissão de dados assíncrona é preferida sobre a síncrona.

Considere uma ligação *Message Queue (MQ)* quando esses fatores são aplicáveis:

- É necessário acessar um sistema de mensagens do WebSphere MQ e é necessário usar as funções nativas do MQ.
- Os serviços estão fracamente acoplados.
- A confiabilidade é mais importante do que o desempenho; ou seja, a transmissão de dados assíncrona é preferida sobre a síncrona.

Considere uma ligação *MQ JMS* quando esses fatores forem aplicáveis:

- É necessário acessar um sistema de mensagens do WebSphere MQ, mas pode fazê-lo em um contexto de JMS, isto é, o subconjunto de funções do JMS é suficiente para seu aplicativo.
- Os serviços estão fracamente acoplados.
- A confiabilidade é mais importante do que o desempenho; ou seja, a transmissão de dados assíncrona é preferida sobre a síncrona.

Ligações SCA

Uma ligação de Service Component Architecture (SCA) permite que um serviço se comunique com outros serviços em outros módulos. Uma importação com uma ligação SCA permite acessar um serviço em outro módulo SCA. Uma exportação com uma ligação SCA permite oferecer um serviço para outros módulos.

Você pode usar o IBM Integration Designer para gerar e configurar ligações do SCA em importações e exportações nos módulos do SCA.

Se os módulos estiverem em execução no mesmo servidor ou forem implementados no mesmo cluster, uma ligação de SCA será a ligação mais fácil e mais rápida a usar.

Depois que o módulo que contém a ligação SCA é implementado no servidor, é possível usar o console administrativo para visualizar informações sobre a ligação ou, no caso de uma ligação de importação, para alterar propriedades selecionadas da ligação.

Ligações de Serviço da Web

Uma ligação de serviços da web é o meio de transmitir mensagens de um componente Service Component Architecture (SCA) para um serviço da web (e vice-versa).

Visão geral das ligações de serviço da Web:

Uma ligação de importação do serviço da web permite chamar um serviço da web externo de seus componentes Service Component Architecture (SCA). Uma ligação de exportação de serviço da web permite expor seus componentes SCA a clientes como serviços da web.

Com uma ligação de serviço da web, você acessa serviços externos usando mensagens SOAP e qualidades de serviço (QoS) interoperáveis.

Você usa o Integration Designer para gerar e configurar ligações de serviço da web em importações e exportações em módulos SCA. Os tipos de ligações de serviços da web a seguir estão disponíveis:

- SOAP1.2/HTTP e SOAP1.1/HTTP

Essas ligações são baseadas em Java API for XML Web Services (JAX-WS), uma API de programação Java para criar serviços da web.

- Use o SOAP1.2/HTTP se seu serviço da web estiver em conformidade com a especificação SOAP 1.2.
- Use o SOAP1.1/HTTP se seu serviço da web estiver em conformidade com a especificação SOAP 1.1.

Importante: Ao implementar um aplicativo com uma ligação de serviço da web (JAX-WS), o servidor de destino não deve ter a opção **Iniciar componentes conforme necessário** selecionada. Acesse “Verificando a Configuração do Servidor” na página 83 para obter detalhes.

Quando você seleciona uma destas ligações, é possível enviar anexos com suas mensagens SOAP.

As ligações de serviço da web funcionam com mensagens SOAP padrão. Usando uma das ligações JAX-WS de serviço da web, no entanto, é possível customizar a forma como as mensagens SOAP são analisadas ou gravadas. Por exemplo, é possível manipular elementos não padrão em mensagens SOAP ou aplicar o processamento adicional na mensagem SOAP. Quando você configura a ligação, especifica um manipulador de dados customizado que executa esse processamento na mensagem SOAP.

É possível usar conjuntos de políticas com uma ligação de serviço da web (JAX-WS). Um conjunto de políticas é uma coleção de tipos de política, cada um dos quais fornece uma qualidade de serviço. Por exemplo, o conjunto de políticas WSAddressing fornece uma forma de transporte neutro para endereçar uniformemente serviços da web e mensagens. Você pode usar o Integration Designer para selecionar o conjunto de políticas para a ligação.

Nota: Se desejar usar um conjunto de políticas Security Assertion Markup Language (SAML), você deverá executar alguma configuração adicional, conforme descrito em “Importando Conjuntos de Políticas SAML” na página 80.

- SOAP1.1/HTTP

Use esta ligação se desejar criar serviços da web que usam uma mensagem codificada por SOAP baseada em Java API for XML-based RPC (JAX-RPC).

- SOAP1.1/JMS

Use esta ligação para enviar ou receber mensagens SOAP usando um destino de Sistema de Mensagens Java (JMS).

Independentemente do transporte (HTTP ou JMS) que é usado para transmitir a mensagem SOAP, as ligações de serviço da web sempre manipulam interações de solicitação/resposta de forma síncrona. O encadeamento que faz a chamada no provedor de serviços é bloqueado até que uma resposta seja recebida do provedor. Consulte “Chamada Síncrona” para obter informações adicionais sobre esse estilo de chamada.

Importante: As seguintes combinações de ligações de serviço da web não podem ser usadas nas exportações no mesmo módulo. Se precisar expor componentes usando mais de uma destas ligações de exportação, você precisará ter cada um em um módulo separado e, em seguida, conectar esses módulos em seus componentes usando a ligação SCA:

- SOAP 1.1/JMS e SOAP 1.1/HTTP usando JAX-RPC
- SOAP 1.1/HTTP usando JAX-RPC e SOAP 1.1/HTTP usando JAX-WS
- SOAP 1.1/HTTP usando JAX-RPC e SOAP 1.2/HTTP usando JAX-WS

Após o módulo SCA que contém a ligação de serviço da web ser implementado no servidor, é possível usar o console administrativo para visualizar informações sobre a ligação ou alterar as propriedades selecionadas da ligação.

Nota: Os serviços da Web permitem que aplicativos interoperem usando descrições padrão de serviços e formatos padrão para as mensagens que eles trocam. Por exemplo, as ligações de importação e exportação de serviço da web podem interoperar com serviços que são implementados usando Web Services Enhancements (WSE) Versão 3.5 e Windows Communication Foundation (WCF) Versão 3.5 para Microsoft .NET. Ao interoperar com tais serviços, você deve assegurar que:

- O arquivo Web Services Description Language (WSDL) que é usado para acessar uma exportação de serviço da web inclui um valor de ação SOAP não vazio para cada operação na interface.
- O cliente de serviço da web configura o cabeçalho SOAPAction ou o cabeçalho wsa:Action ao enviar mensagens a uma exportação de serviço da web.

Propagação de Cabeçalho SOAP:

Ao manipular mensagens SOAP, pode ser necessário acessar informações de determinados cabeçalhos SOAP em mensagens que são recebidas, assegurar que as mensagens com cabeçalhos SOAP sejam enviadas com valores específicos ou permitir que os cabeçalhos SOAP passem por meio de um módulo.

Ao configurar uma ligação de serviço da web no Integration Designer, é possível indicar que deseja que os cabeçalhos SOAP sejam propagados.

- Quando as solicitações são recebidas em uma exportação ou respostas são recebidas em uma importação, as informações do cabeçalho SOAP podem ser acessadas, permitindo que a lógica no módulo seja baseada em valores de cabeçalho e permitindo que esses cabeçalhos sejam modificados.
- Quando solicitações são enviadas de uma exportação ou respostas são enviadas de uma importação, cabeçalhos SOAP podem ser incluídos nessas mensagens.

O formato e a presença dos cabeçalhos SOAP propagados podem ser afetados pelos conjuntos de política configurados na importação ou exportação, conforme explicado em Tabela 31 na página 77.

Para configurar a propagação de cabeçalhos SOAP para uma importação ou exportação, você seleciona (a partir da visualização Propriedades do Integration Designer) a guia **Propagar Cabeçalho do Protocolo** e seleciona as opções requeridas.

Cabeçalho WS-Addressing

O cabeçalho WS-Addressing pode ser propagado pela ligação de serviço da web (JAX-WS).

Quando você propagar o cabeçalho WS-Addressing, esteja ciente das seguintes informações:

- Se você ativar a propagação para o cabeçalho WS-Addressing, o cabeçalho será propagado no módulo nas seguintes circunstâncias:
 - Quando as solicitações são recebidas em uma exportação
 - Quando as respostas são recebidas em uma importação
- O cabeçalho WS-Addressing não é propagado para mensagens de saída a partir do IBM Business Process Manager (ou seja, o cabeçalho não é propagado quando as solicitações são enviadas a partir de uma importação ou quando as respostas são enviadas a partir da exportação).

Cabeçalho WS-Security

O cabeçalho WS-Security pode ser propagado pela ligação de serviço da web (JAX-WS) e a ligação de serviço da web (JAX-RPC).

A especificação WS-Security de serviços da web descreve aprimoramentos no sistema de mensagens SOAP para fornecer uma qualidade de proteção por meio da integridade da mensagem, da confidencialidade da mensagem e da autenticação de mensagem única. Esses mecanismos podem ser usados para acomodar uma grande variedade de modelos de segurança e tecnologias de criptografia.

Ao propagar o cabeçalho WS-Security, observe as seguintes informações:

- Se você ativar a propagação para o cabeçalho WS-Security, o cabeçalho será propagado por meio do módulo nas seguintes circunstâncias:
 - Quando as solicitações são recebidas em uma exportação
 - Quando solicitações são enviadas a partir de uma importação
 - Quando as respostas são recebidas em uma importação
- O cabeçalho *não* será, por padrão, propagado quando as respostas forem enviadas a partir da exportação. No entanto, se você configurar a propriedade JVM **WSSECURITY.ECHO.ENABLED** como **true**, o cabeçalho será propagado quando as respostas forem enviadas a partir da exportação. Nesse caso, se o cabeçalho WS-Security no caminho da solicitação não for modificado, os cabeçalhos WS-Security poderão ser ecoados automaticamente das solicitações para as respostas.
- O formato exato da mensagem SOAP enviada a partir de uma importação para uma solicitação ou a partir de uma exportação para uma resposta poderá não corresponder exatamente à mensagem SOAP que foi recebida originalmente. Por esse motivo, qualquer assinatura digital deve ser assumida para se tornar inválida. Se uma assinatura digital for requerida em mensagens que são enviadas, ela deverá ser estabelecida usando o conjunto de políticas de segurança apropriado e cabeçalhos WS-Security relacionados à assinatura digital em mensagens recebidas devem ser removidos dentro do módulo.

Para propagar o cabeçalho WS-Security, você deve incluir o esquema WS-Security com o módulo aplicativo. Consulte “Incluindo o Esquema WS-Security em um Módulo Aplicativo” na página 77 para conhecer o procedimento para incluir o esquema.

Como os Cabeçalhos são Propagados

A forma com que os cabeçalhos são propagados depende da configuração da política de segurança na ligação de importação ou exportação, conforme mostrado no Tabela 31 na página 77:

Tabela 60. Como Cabeçalhos de Segurança são Transmitidos

	Ligação de exportação sem política de segurança	Ligação de exportação com política de segurança
Ligação de importação sem política de segurança	<p>Os cabeçalhos de segurança são transmitidos no estado em que se encontram através do módulo. Eles não são descriptografados.</p> <p>Os cabeçalhos são enviados no mesmo formato em que foram recebidos.</p> <p>A assinatura digital pode se tornar inválida.</p>	<p>Os cabeçalhos de segurança são descriptografados e passados pelo módulo com verificação de assinatura e autenticação.</p> <p>Os cabeçalhos descriptografados são enviados.</p> <p>A assinatura digital pode se tornar inválida.</p>
Ligação de importação com política de segurança	<p>Os cabeçalhos de segurança são transmitidos no estado em que se encontram através do módulo. Eles não são descriptografados.</p> <p>Os cabeçalhos não devem ser propagados para a importação. Caso contrário, ocorre um erro devido à duplicação.</p>	<p>Os cabeçalhos de segurança são descriptografados e passados pelo módulo com verificação de assinatura e autenticação.</p> <p>Os cabeçalhos não devem ser propagados para a importação. Caso contrário, ocorre um erro devido à duplicação.</p>

Configure os conjuntos de políticas apropriados nas ligações de exportação e importação, pois isso isola o solicitante de serviços das alterações na configuração ou requisitos de QoS do provedor de serviços. Tendo cabeçalhos SOAP padrão visíveis em um módulo, eles podem, então, ser usados para influenciar o processamento (por exemplo, criação de log e rastreamento) no módulo. Propagar os cabeçalhos SOAP em um módulo a partir de uma mensagem recebida para uma mensagem enviada significa que os benefícios de isolamento do módulo são reduzidos.

Cabeçalhos padrão, como cabeçalhos WS-Security, não devem ser propagados em uma solicitação para uma importação ou resposta para uma exportação quando a importação ou exportação possui um conjunto de política associado que normalmente resultaria na geração desses cabeçalhos. Caso contrário, ocorrerá um erro devido a uma duplicação dos cabeçalhos. Em vez disso, os cabeçalhos devem ser explicitamente removidos ou a ligação de importação ou exportação deve ser configurada para evitar a propagação dos cabeçalhos de protocolo.

Acessando Cabeçalhos SOAP

Quando uma mensagem que contém os cabeçalhos SOAP é recebida de uma importação ou exportação de serviço da web, os cabeçalhos são posicionados na seção do cabeçalho do objeto de mensagem de serviço (SMO). É possível acessar as informações de cabeçalho, conforme descrito em "Acessando Informações de Cabeçalho SOAP no SMO".

Incluindo o Esquema WS-Security em um Módulo Aplicativo

O procedimento a seguir descreve as etapas para incluir o esquema no módulo aplicativo:

- Se o computador no qual o Integration Designer está em execução tem acesso à Internet, execute as seguintes etapas:
 1. Na perspectiva de Integração de Negócios, selecione **Dependências** para seu projeto.
 2. Expanda **Recursos Predefinidos** e selecione **Arquivos de Esquema WS-Security 1.0** ou **Arquivos de Esquema WS-Security 1.1** para importar o esquema em seu módulo.
 3. Limpe e reconstrua o projeto.

- Se um computador no qual o Integration Designer está executando não tiver acesso à Internet, você poderá fazer download do esquema para um segundo computador que tenha acesso à Internet. É possível, então, copiá-lo no computador no qual o Integration Designer está em execução.

1. No computador que possui acesso à Internet, faça download do esquema remoto:

a. Clique em **Arquivo > Importar > Integração de Negócios > WSDL e XSD**.

b. Selecione **WSDL Remoto** ou **Arquivo XSD**.

c. Importe os esquemas a seguir:

`http://www.w3.org/2003/05/soap-envelope/`

`http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd`

`http://www.w3.org/TR/xmlsig-core/xmlsig-core-schema.xsd`

2. Copie os esquemas no computador que não possui acesso à Internet.

3. No computador que não possui acesso à Internet, importe o esquema:

a. Clique em **Arquivo > Importar > Integração de Negócios > WSDL e XSD**.

b. Selecione **WSDL Local** ou **Arquivo XSD**.

4. Altere os locais de esquema para `oasis-wss-wssecurity-secext-1.1.xsd`:

a. Abra o esquema em `workplace_location/module_name/StandardImportFilesGen/oasis-wss-wssecurity-secext-1.1.xsd`.

b. Altere:

```
<xs:import namespace='http://www.w3.org/2003/05/soap-envelope'
schemaLocation='http://www.w3.org/2003/05/soap-envelope/' />
```

para:

```
<xs:import namespace='http://www.w3.org/2003/05/soap-envelope'
schemaLocation='../w3/_2003/_05/soap_envelope.xsd' />
```

c. Altere:

```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
schemaLocation='http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd' />
```

para:

```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
schemaLocation='../w3/tr/_2002/rec_xmlenc_core_20021210/xenc-schema.xsd' />
```

5. Altere o local de esquema para `oasis-200401-wss-wssecurity-secext-1.0.xsd`:

a. Abra o esquema em `workplace_location/module_name/standardimportfilesgen/oasis-200401-wss-wssecurity-secext-1.0.xsd`.

b. Altere:

```
<xsd:import namespace="http://www.w3.org/2000/09/xmlsig#"
schemaLocation="http://www.w3.org/TR/xmlsig-core/xmlsig-core-schema.xsd"/>
```

para:

```
<xsd:import namespace="http://www.w3.org/2000/09/xmlsig#"
schemaLocation="../w3/tr/_2002/rec_xmlsig_core_20020212/xmlsig-core-schema.xsd"/>
```

6. Limpe e reconstrua o projeto.

Propagação do Cabeçalho de Transporte:

Ao manipular mensagens SOAP, pode ser necessário acessar informações de determinados cabeçalhos de transporte em mensagens que são recebidas, assegurar que as mensagens com cabeçalhos transport sejam enviadas com valores específicos ou permitir que os cabeçalhos de transporte passem por meio de um módulo.

Ao configurar uma ligação de serviço da web no Integration Designer, é possível indicar que deseja que os cabeçalhos de transporte sejam propagados.

- Quando solicitações são recebidas em uma exportação ou respostas são recebidas em uma importação, as informações do cabeçalho de transporte podem ser acessadas, permitindo que a lógica no módulo seja baseada nos valores de cabeçalho e permitindo que esses cabeçalhos sejam modificados.
- Quando respostas são enviadas a partir de uma exportação ou solicitações são enviadas a partir de uma importação, os cabeçalhos de transporte podem ser incluídos nessas mensagens.

Especificando a Propagação de Cabeçalhos

Para configurar a propagação de cabeçalhos de transporte para uma importação ou exportação, execute as seguintes etapas:

1. Na visualização Propriedades do Integration Designer, selecione **Ligação > Propagação**.
2. Configure a opção de propagação de cabeçalho de transporte que você requer.

Nota: A propagação do cabeçalho de transporte é desativada por padrão e pode ser implementada apenas em um ambiente de tempo de execução Versão 7.0.0.3 (ou posterior). Além disso, observe que, para a Versão 7.0.0.3, a propagação de cabeçalho de transporte é limitada a cabeçalhos de transporte HTTP apenas.

Se você ativar a propagação de cabeçalhos de transporte, os cabeçalhos serão propagados em um módulo a partir de mensagens recebidas e, se você não remover explicitamente os cabeçalhos, eles serão usados nas chamadas subsequentes no mesmo encadeamento.

Nota: Os cabeçalhos de transporte não podem ser propagados ao usar a ligação de serviço da web (JAX-RPC).

Acessando as Informações de Cabeçalho

Quando a propagação de cabeçalho de transporte é ativada para mensagens recebidas, todos os cabeçalhos de transporte (incluindo cabeçalhos definidos pelo cliente) ficam visíveis no objeto de mensagem do serviço (SMO). É possível configurar os cabeçalhos com valores diferentes ou criar novos. Observe, no entanto, que não há verificação ou validação dos valores configurados e qualquer cabeçalho impróprio ou incorreto pode causar problemas de tempo de execução de serviço da web.

Considere as seguintes informações sobre a configuração de cabeçalhos HTTP:

- Quaisquer mudanças nos cabeçalhos reservados para o mecanismo de serviço da web não serão aceitas na mensagem de saída. Por exemplo, a versão de HTTP ou método, os cabeçalhos Content-Type, Content-Length e SOAPAction serão reservados para o mecanismo do serviço da web.
- Se o valor do cabeçalho for um número, o número (em vez da sequência) deverá ser configurado diretamente. Por exemplo, use **Max-Forwards = 5** (em vez de **Max-Forwards = Max-Forwards: 5**) e **Age = 300** (em vez de **Age = Age: 300**).
- Se a mensagem de solicitação for menor que 32 KB de tamanho, o mecanismo de serviço da web removerá o cabeçalho Transfer-Encoding e no lugar configurará o cabeçalho Content-Length para o tamanho fixo da mensagem.
- O Content-language é reconfigurado por was.channel.http no caminho de resposta.
- Uma configuração inválida para resultados de Upgrade em um erro 500.
- Os cabeçalhos a seguir anexam o valor reservado pelo mecanismo de serviço da web para as configurações do cliente:
 - User-Agent
 - Cache-Control
 - Pragma
 - Accept
 - Conexão

É possível acessar as informações do cabeçalho de uma das maneiras a seguir:

- Usando uma primitiva de mediação para acessar as estruturas de SMO
Consulte os links “Informações Relacionadas” para localizar informações sobre como usar primitivas de mediação.
- Usando o serviço de contexto SPI

O código de amostra a seguir lê os cabeçalhos de transporte HTTP do serviço de contexto:

```
HeadersType headerType = ContextService.INSTANCE.getHeaders();
HTTPHeaderType httpHeaderType = headerType.getHTTPHeader();
List HTTPHeader httpHeaders = httpHeaderType.getHeader();
if(httpHeaders!=null){
    for(HTTPHeader httpHeader: httpHeaders){
        String httpHeadername = httpHeader.getName();
        String httpHeaderValue = httpHeader.getValue();
    }
}
List PropertyType properties = headerType.getProperties();
if(properties!=null){
    for(PropertyType property: properties){
        String propertyName = property.getName();
        String propertyValue = property.getValue().toString();
    }
}
```

Resolução de problemas

Se você encontrar problemas ao enviar os cabeçalhos revisados, poderá interceptar a mensagem TCP/IP usando ferramentas tais como o Monitor TCP/IP no Integration Designer. Você acessa o Monitor TCP/IP selecionando **Executar/Depurar > Monitor TCP/IP** na página Preferências.

Também é possível visualizar os valores de cabeçalho usando o rastreamento do mecanismo do JAX-WS:
org.apache.axis2.*=all: com.ibm.ws.websvcs.*=all:

Trabalhando com Ligações de Serviço da Web (JAX-WS):

Ao usar as ligações de serviço da web (JAX-WS) com seus aplicativos, é possível incluir uma qualidade de serviço (QOS) Security Assertion Markup Language (SAML) na ligação. Você deve primeiro usar o console administrativo para importar o conjunto de políticas. Também é possível usar o console administrativo para certificar-se de que o servidor esteja corretamente configurado para uso com a ligação de serviço da web (JAX-WS).

Importando Conjuntos de Políticas SAML:

Security Assertion Markup Language (SAML) é um padrão OASIS baseado em XML para trocar informações de identidade do usuário e de atributos de segurança. Quando configurar uma ligação de serviço da web (JAX-WS) no Integration Designer, é possível especificar um conjunto de políticas SAML. Primeiro use o console administrativo do IBM Business Process Manager para tornar os conjuntos de políticas SAML disponíveis para que eles possam ser importados no Integration Designer.

Os conjuntos de políticas SAML estão geralmente localizados no diretório de configuração do perfil:

profile_root/config/templates/PolicySets

Antes de iniciar este procedimento, verifique se os seguintes diretórios (que contêm os conjuntos de políticas) estão localizados no diretório de configuração do perfil:

- SAML11 Bearer WSHTTPS padrão
- SAML20 Bearer WSHTTPS padrão
- SAML11 Bearer WSSecurity padrão

- SAML20 Bearer WSSecurity padrão
- SAML11 HoK Public WSSecurity padrão
- SAML20 HoK Public WSSecurity padrão
- SAML11 HoK Symmetric WSSecurity padrão
- SAML20 HoK Symmetric WSSecurity padrão
- Padrão WSHTTPS de Nome do Usuário

Se os diretórios não estiverem no diretório de configuração do perfil, copie-os nesse diretório a partir do seguinte local:

`app_server_root/profileTemplates/default/documents/config/templates/PolicySets`

Você pode importar os conjuntos de política no console administrativo, selecionar aqueles que deseja disponibilizar no Integration Designer e salvar um arquivo .zip para cada um dos conjuntos de política em um local que seja acessível ao Integration Designer.

1. Importe os conjuntos de políticas seguindo estas etapas:
 - a. No console administrativo, clique em **Serviços > Conjuntos de Políticas > Conjuntos de Políticas de Aplicativo**.
 - b. Clique em **Importar > A Partir do Repositório Padrão**.
 - c. Selecione os conjuntos de políticas padrão SAML e clique em **OK**.
2. Exporte os conjuntos de políticas para que possam ser usados pelo Integration Designer:
 - a. Na página Conjuntos de Políticas de Aplicativo, selecione o conjunto de políticas SAML que deseja exportar e clique em **Exportar**.

Nota: Se a página Conjuntos de Políticas de Aplicativo não for exibida atualmente, clique em **Serviços > Conjuntos de Políticas > Conjuntos de Políticas de Aplicativo** a partir do console administrativo.

- b. Na próxima página, clique no link do arquivo .zip para o conjunto de políticas.
- c. Na janela Download de Arquivo, clique em **Salvar** e indique um local que esteja acessível para o Integration Designer.
- d. Clique em **Voltar**.
- e. Conclua as etapas 2a na página 81 a 2d na página 81 para cada conjunto de políticas que deseja exportar.

Os conjuntos de políticas SAML são salvos nos arquivos .zip e estão prontos para serem importados no Integration Designer.

Importe os conjuntos de políticas no Integration Designer, conforme descrito no tópico “Conjuntos de Políticas”.

Chamando Serviços da Web que Requerem Autenticação HTTP Básica:

A autenticação básica de HTTP utiliza um nome de usuário e uma senha para autenticar um cliente de serviço em um terminal seguro. É possível configurar a autenticação básica de HTTP ao enviar ou receber solicitações de serviços da Web.

Você pode configurar a autenticação básica de HTTP para receber solicitações de serviço da web configurando a ligação de exportação da API Java para Serviços da Web XML (JAX-WS), conforme descrito em Criando e designando funções de segurança a exportações de serviço da web.

Autenticação básica de HTTP pode ser ativada para as solicitações de serviço da Web que são enviadas por uma ligação de importação JAX-WS em uma de duas maneiras:

- Ao configurar a ligação de importação em um módulo SCA, é possível selecionar o conjunto de políticas de autenticação HTTP fornecido denominado BPMHTTPBasicAuthentication (que é fornecido com a ligação de importação de serviço da Web (JAX-WS)) ou qualquer outro conjunto de políticas que inclua a política HTTPTransport.
- Ao construir o módulo SCA, é possível usar os recursos de fluxo de mediação para criar dinamicamente um novo cabeçalho de autenticação HTTP e especificar as informações de nome de usuário e senha no cabeçalho.

Nota: O conjunto de políticas tem precedência sobre o valor especificado no cabeçalho. Se você deseja usar o valor configurado no cabeçalho de autenticação HTTP no tempo de execução, não conecte um conjunto de políticas que inclua a política HTTPTransport. Especificamente, não use o conjunto de políticas BPMHTTPBasicAuthentication padrão e, se tiver definido um conjunto de políticas, certifique-se de que ele exclua a política HTTPTransport.

Para obter mais informações sobre os conjuntos de políticas de serviço da web e as ligações de diretrizes e como são usados, consulte Conjuntos de políticas de serviços da web do Centro de Informações do WebSphere Application Server.

- Para usar o conjunto de políticas fornecido, execute as etapas a seguir:
 1. Opcional: No console administrativo, crie uma ligação de diretrizes gerais do cliente ou edite uma existente que inclua a política HTTPTransport com os valores de ID de usuário e senha necessários.
 2. No IBM Integration Designer, gere uma ligação de importação de serviço da Web (JAX-WS) e anexe o conjunto de políticas BPMHTTPBasicAuthentication.
 3. Execute *uma* das etapas a seguir:
 - No IBM Integration Designer, nas propriedades de ligação de importação de serviço da Web (JAX-WS), especifique o nome de uma ligação de política geral de cliente existente que inclua a política HTTPTransport.
 - Após implementar o módulo SCA, use o console administrativo para selecionar uma ligação de diretrizes do cliente existente ou crie uma nova ligação de diretrizes do cliente e associe-a à ligação de importação.
 4. Opcional: No console administrativo do servidor de processo, edite a ligação do conjunto de políticas selecionada para especificar o ID e a senha necessários.
- Para especificar o nome de usuário e a senha no cabeçalho de autenticação HTTP, execute um dos seguintes conjuntos de etapas:
 - Use a primitiva de mediação do Configurador de Cabeçalho HTTP no IBM Integration Designer para criar o cabeçalho de autenticação HTTP e especifique o nome do usuário e a senha.
 - Se for necessária lógica adicional, use o código Java em uma primitiva de mediação customizada (conforme mostrado no exemplo a seguir) para:
 1. Criar um cabeçalho de autenticação HTTP.
 2. Especificar as informações de nome de usuário e senha.
 3. Incluir o novo cabeçalho de autenticação HTTP em HTTPControl.
 4. Configurar o HTTPControl atualizado de volta no serviço de Contexto.

```
//Get the HeaderInfoType from contextService
ContextService contextService = (ContextService) ServiceManager.INSTANCE
.locateService("com/ibm/bpm/context/ContextService");
HeaderInfoType headers = contextService.getHeaderInfo();
if(headers == null){
    headers = ContextObjectFactory.eINSTANCE.createHeaderInfoType();
}
//Get the HTTP header and HTTP Control from HeaderInfoType
HTTPHeaderType httpHeaderType = headers.getHTTPHeader();
HTTPControl cp = httpHeaderType.getControl();
HeadersFactory factory = HeadersFactory.eINSTANCE;
if(cp == null){
    cp = factory.createHTTPControl();
}
```

```

}
//Create new HTTPAuthentication and set the HTTPCredentials
HTTPAuthentication authorization = factory.createHTTPAuthentication();
HTTPCredentials credentials = factory.createHTTPCredentials();
authorization.setAuthenticationType(HTTPAuthenticationType.BASIC_LITERAL);
credentials.setUserId("USERNAME");
credentials.setPassword("PASSWORD");
authorization.setCredentials(credentials);
cp.setAuthentication(authorization);
httpHeaderType.setControl(cp);
// Set header info back to the current execution context.
contextService.setHeaderInfo(headers);

```

Verificando a Configuração do Servidor:

Ao implementar um aplicativo com uma ligação de serviço da web (JAX-WS), você deve certificar-se de que o servidor no qual o aplicativo está implementado não tenha a opção **Iniciar componentes como necessário** selecionada.

É possível verificar se esta opção está selecionada executando as seguintes etapas a partir do console administrativo:

1. Clique em **Servidores > Tipos de Servidor > WebSphere Application Servers**.
2. Clique no nome do servidor.
3. Na guia Configuração, determine se **Iniciar Componentes Conforme Necessário** está selecionado.
4. Execute uma das etapas a seguir:
 - Se **Iniciar Componentes Conforme Necessário** estiver selecionado, remova a seleção e, em seguida, clique em **Aplicar**.
 - Se **Iniciar Componentes Conforme Necessário** não estiver selecionado, clique em **Cancelar**.

Anexos em mensagens SOAP:

É possível enviar e receber mensagens SOAP que incluem dados binários (como arquivos PDF ou imagens JPEG) como anexos. Anexos podem ser *referenciados* (ou seja, representados como partes da mensagem explicitamente na interface de serviço) ou *não referenciados* (em que números e tipos arbitrários de anexos podem ser incluídos).

Um anexo referenciado pode ser representado de uma das maneiras a seguir:

- Anexos MTOM usam a codificação especificada do SOAP Message Transmission Optimization Mechanism (<http://www.w3.org/TR/soap12-mtom/>). Anexos MTOM são ativados por meio de uma opção de configuração nas ligações de importação e exportação e são a maneira recomendada para codificar anexos para novos aplicativos.
- Como um elemento wsi:swaRef-typed no esquema de mensagem
Anexos definidos usando o tipo wsi:swaRef em conformidade com o Web Services Interoperability Organization (WS-I) *Attachments Profile Versão 1.0* (<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), que define como elementos de mensagem são relacionados a partes MIME.
- Como uma parte da mensagem de nível superior, usando um tipo de esquema binário
Anexos representados como partes da mensagem de nível superior em conformidade com a especificação *Mensagens SOAP com Anexos* (<http://www.w3.org/TR/SOAP-attachments>).
Anexos representados como partes da mensagem de nível superior também podem ser configurados para assegurar que o documento WSDL e as mensagens produzidos pela ligação estejam em conformidade com o *WS-I Attachments Profile Versão 1.0* e o *WS-I Basic Profile Versão 1.1* (<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>).

Um anexo não referenciado é transportado em uma mensagem SOAP sem qualquer representação no esquema de mensagem.

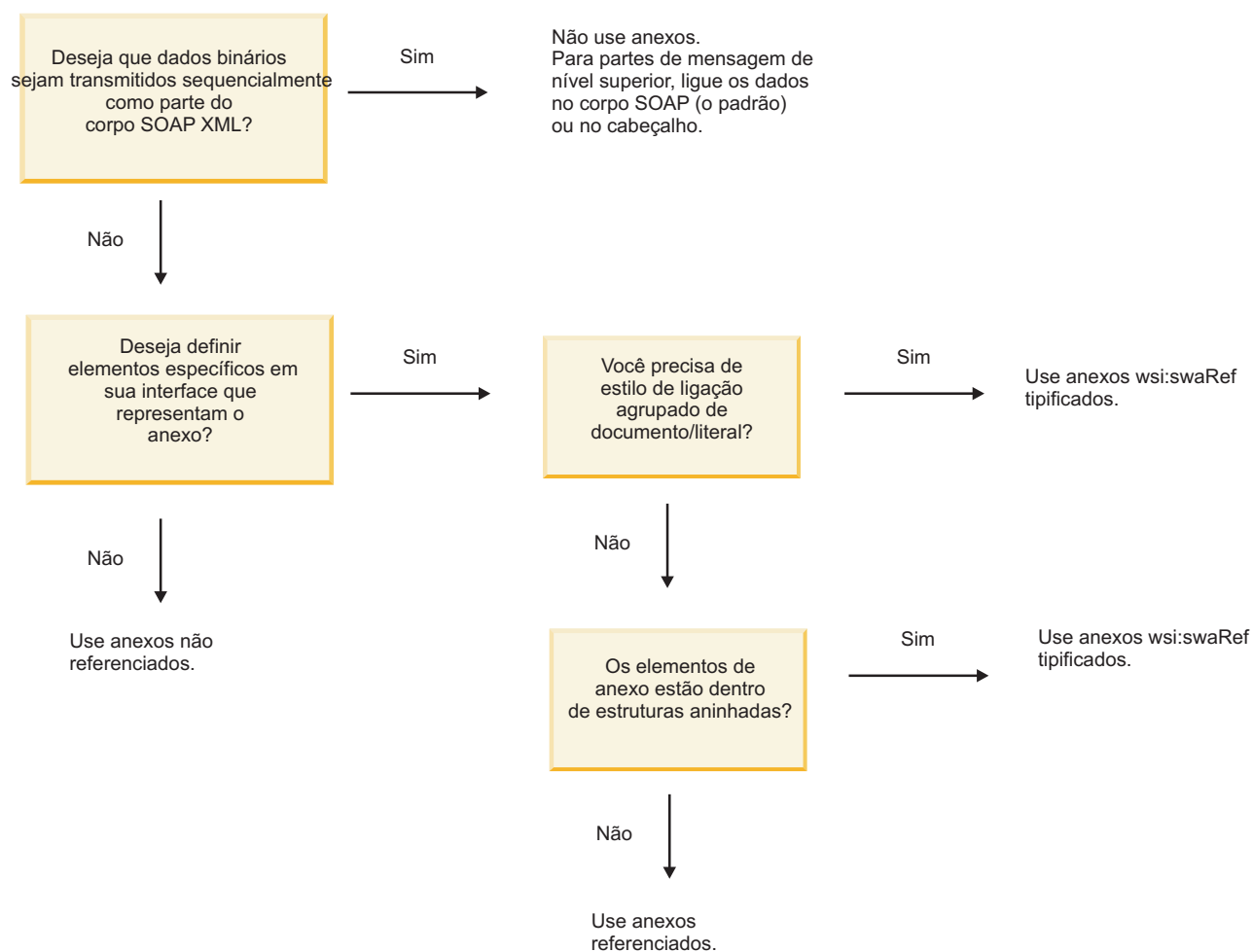
Em todos os casos, exceto anexos MTOM, a ligação WSDL SOAP deve incluir uma ligação MIME para anexos a serem usados, e o tamanho máximo dos anexos não deve exceder 20 MB.

Nota: Para enviar ou receber mensagens SOAP com anexos, você deve usar uma das ligações de serviços da web baseadas em Java API for XML Web Services (JAX-WS).

Como Escolher o Estilo Adequado de Anexo:

Ao projetar uma nova interface de serviço que inclui dados binários, considere como esses dados binários são transportados nas mensagens SOAP que são enviadas e recebidas pelo serviço.

O Message Transmission Optimization Mechanism (MTOM) deve ser usado para anexos se o aplicativo de serviços da web suportá-lo. Se não, o seguinte diagrama mostra como outros estilos de anexo são escolhidos. Use o seguinte conjunto de perguntas para determinar o estilo de anexo apropriado:



Anexos MTOM: Partes da Mensagem em Nível Superior:

É possível enviar e receber mensagens de serviço da web que incluem anexos SOAP Message Transmission Optimization Mechanism (MTOM). Em uma mensagem SOAP MIME com múltiplas partes, o corpo SOAP é a primeira parte da mensagem e o anexo ou anexos estão em partes subsequentes.

Ao enviar ou receber um anexo referenciado em uma mensagem SOAP, os dados binários que compõem o anexo (que é geralmente muito grande) são mantidos separadamente do corpo da mensagem SOAP

para que não precisem ser analisados como XML. Isso resulta em processamento mais eficiente do que se os dados binários fossem mantidos dentro de um elemento XML.

A seguir há uma amostra de uma mensagem SOAP MTOM:

```
... outros cabeçalhos de transporte ...
Content-Type: multipart/related; boundary=MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812;
type="application/xop+xml"; start="
<0.urn:uuid:0FE43E4D025F0BF3DC11582467646813@apache.org>";
start-info="text/xml"; charset=UTF-8

--MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812
content-type: application/xop+xml; charset=UTF-8; type="text/xml";
content-transfer-encoding: binary
content-id:
  <0.urn:uuid:0FE43E4D025F0BF3DC11582467646813@apache.org>

<?xml version="1.0" encoding="UTF-8"?>
  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header/>
    <soapenv:Body>
      <sendImage xmlns="http://org.apache.axis2/jaxws/sample/mtom">
        <input>
          <imageData><xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include"
href="cid:1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org"/></imageData>
        </input>
      </sendImage>
    </soapenv:Body>
  </soapenv:Envelope>
--MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812
content-type: text/plain
content-transfer-encoding: binary
content-id:
  <1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org>

... dados binários vão aqui ...
--MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812--
```

Note que na amostra de MTOM, o tipo de conteúdo para o envelope SOAP é **application/xop+xml** e os dados binários são substituídos por um elemento **xop:Include**, conforme abaixo:

```
<xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include" href="cid:1.urn:uuid:0FE43E4D025F0BF3DC11582467646811@apache.org"/>
```

Processamento de Entrada de Anexos Referenciados

Quando um cliente passa uma mensagem SOAP com um anexo a um componente Service Component Architecture (SCA), a ligação de exportação (JAX-WS) de serviço da web primeiro remove o anexo. Ela, então, analisa a parte SOAP da mensagem e cria um objeto de negócios. Finalmente, a ligação configura o binário de anexo no objeto de negócios.

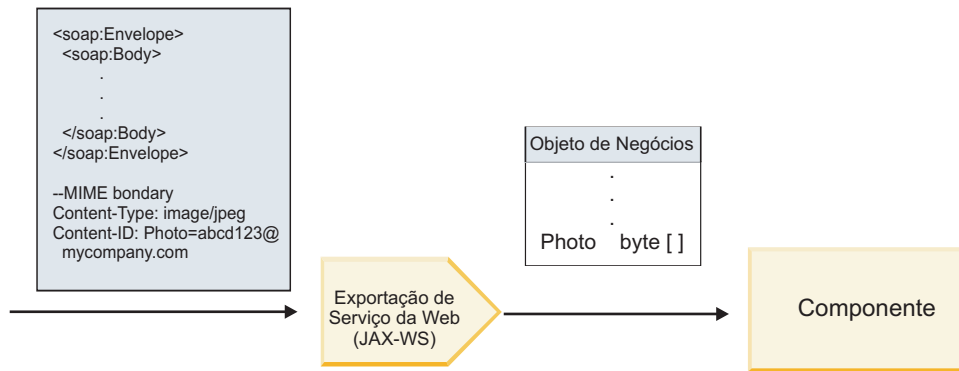


Figura 58. Como a Ligação de Exportação (JAX-WS) de Serviço da Web Processa uma Mensagem SOAP com um Anexo Referenciado

Atributos do Anexo MTOM

- O MTOM pode suportar elementos de anexo dentro de estruturas aninhadas.
- O MTOM somente está disponível para o tipo `base64Binary`.
- O MTOM pode suportar elementos de anexo dentro de estruturas aninhadas, o que significa que o **bodyPath** para anexos do MTOM são o local **xpath** para o elemento no qual o anexo MTOM é retido. A lógica de computação para **bodyPath** é seguir estritamente o esquema para gerar o local **xpath** conforme mostrado nos exemplos abaixo:
 - Para um tipo de não matriz (**maxOccurs** é 1): `/sendImage/input/imageData`
 - Para um tipo de matriz (**maxOccurs** > 1): `/sendImage/input/imageData[1]`
- Tipos de anexo combinados não são suportados, o que significa que se o MTOM está ativado na ligação de importação, o anexo MTOM será gerado. Se o MTOM está desativado ou se o valor de configuração do MTOM é deixado como o valor padrão na ligação de exportação, a mensagem padrão do MTOM não é suportada.

Anexos referenciados: elementos *swaRef*-typed:

Você pode enviar e receber mensagens SOAP que incluem anexos representados na interface de serviço como elementos com tipo *swaRef*.

Um elemento com tipo *swaRef* é definido no Web Services Interoperability Organization (WS-I) *Attachments Profile* Versão 1.0 (<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), o qual define como elementos de mensagem estão relacionados a partes MIME.

Na mensagem SOAP, o corpo SOAP contém um elemento com tipo *swaRef* que identifica o ID de conteúdo do anexo.

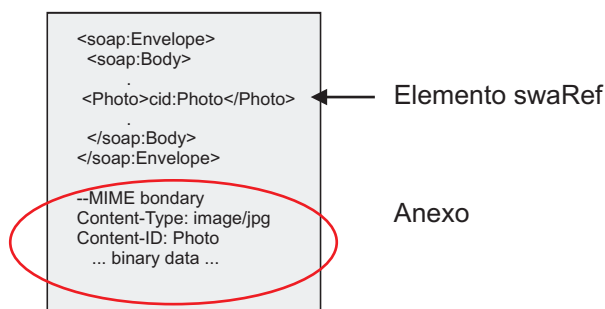


Figura 59. Uma Mensagem SOAP com um Elemento swaRef

O WSDL para essa mensagem SOAP contém um elemento com tipo swaRef dentro de uma parte da mensagem que identifica o anexo.

```
<element name="sendPhoto">
  <complexType>
    <sequence>
      <element name="Photo" type="wsi:swaRef"/>
    </sequence>
  </complexType>
</element>
```

O WSDL também deve conter uma ligação MIME que indica que mensagens MIME multipartes devem ser usadas.

Nota: O WSDL *não* inclui uma ligação MIME para o elemento de mensagem com tipo swaRef específico, porque as ligações MIME se aplicam apenas a partes da mensagem de nível superior.

Os anexos representados como elementos com tipo swaRef podem ser propagados apenas entre os componentes do fluxo de mediação. Se um anexo precisar ser acessado ou propagado para outro tipo de componente, use um componente do fluxo de mediação para mover o anexo para um local que seja acessível para esse componente.

Processamento de Entrada de Anexos

Você pode usar o Integration Designer para configurar uma ligação de exportação para receber o anexo. Você cria um módulo e sua interface e operações associadas, incluindo um elemento de tipo swaRef. Você cria uma ligação de serviço da web (JAX-WS).

Nota: Consulte o tópico “Trabalhando com Anexos” no centro de informações do Integration Designer para obter informações mais detalhadas.

Quando um cliente passa uma mensagem SOAP com um anexo swaRef a um componente Service Component Architecture (SCA), a ligação de exportação (JAX-WS) de serviço da web primeiro remove o anexo. Ela, então, analisa a parte SOAP da mensagem e cria um objeto de negócios. Finalmente, a ligação configura o ID de conteúdo do anexo no objeto de negócios.

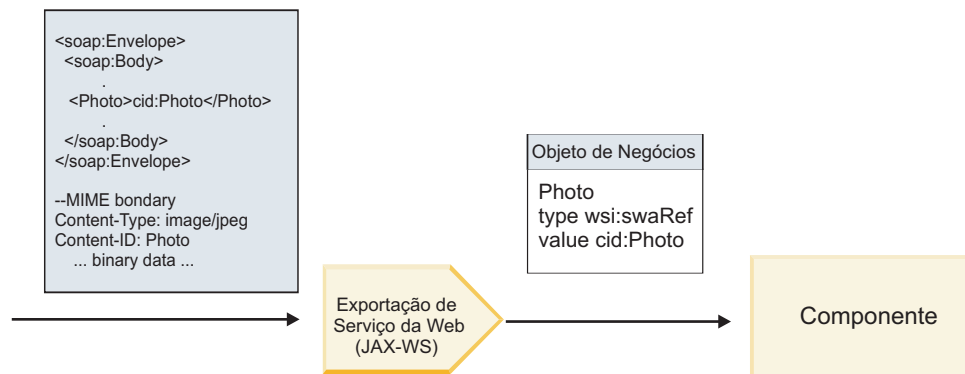


Figura 60. Como a Ligação de Exportação (JAX-WS) de Serviço da Web Processa uma Mensagem SOAP com um Anexo swaRef

Acessando Metadados de Anexo em um Componente do Fluxo de Mediação

Conforme mostrado em Figura 16 na página 88, quando anexos swaRef são acessados por componentes, o identificador do conteúdo do anexo aparece como um elemento de tipo swaRef.

Cada anexo de uma mensagem SOAP também possui um elemento **attachments** correspondente no SMO. Ao usar o tipo swaRef WS-I, o elemento **attachments** inclui o tipo de conteúdo do anexo e o ID do conteúdo bem como os dados binários reais do anexo.

Para obter o valor de um anexo swaRef, é necessário, portanto, obter o valor do elemento com tipo swaRef e, em seguida, localizar o elemento **attachments** com o valor **contentID** correspondente. Observe que o valor **contentID** geralmente tem o prefixo **cid:** removido do valor de swaRef.

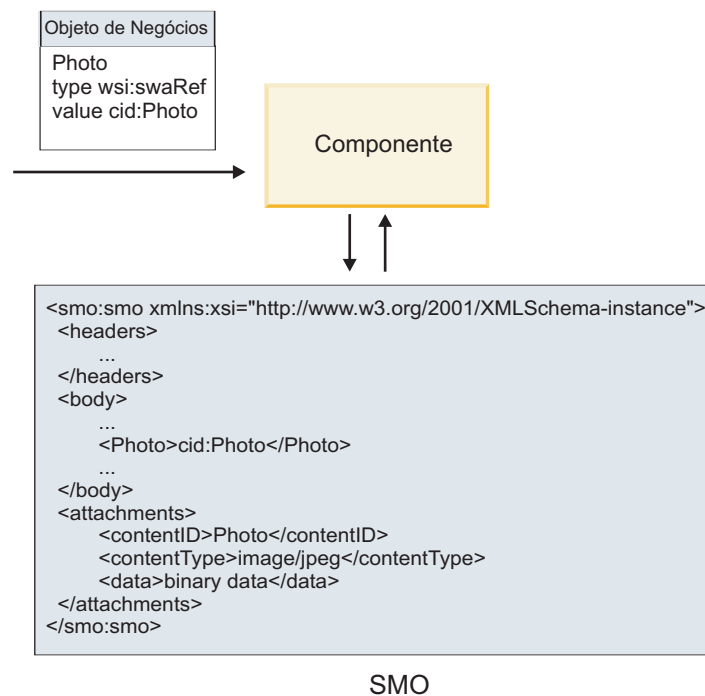


Figura 61. Como Anexos swaRef Aparecem no SMO

Processamento de Saída

Você usa o Integration Designer para configurar uma ligação de importação (JAX-WS) de serviço da web para chamar um serviço da web externo. A ligação de importação é configurada com um documento WSDL que descreve o serviço da web a ser chamado e define o anexo que será passado para o serviço da web.

Quando uma mensagem SCA é recebida por uma ligação de importação (JAX-WS) de serviço da web, os elementos de tipo swaRef são enviados como anexos se a importação estiver ligada a um componente do fluxo de mediação e o elemento de tipo swaRef possuir um elemento correspondente **attachments**.

Para processamento de saída, elementos com tipo swaRef são sempre enviados com seus valores de ID de conteúdo; no entanto, o módulo de mediação deve assegurar que haja um elemento **attachments** correspondente com um valor **contentID** correspondente.

Nota: Para conformidade com o WS-I Attachments Profile, o valor **content ID** deve seguir a "codificação da parte do ID do conteúdo", conforme descrito na seção 3.8 do *WS-I Attachments Profile 1.0*.

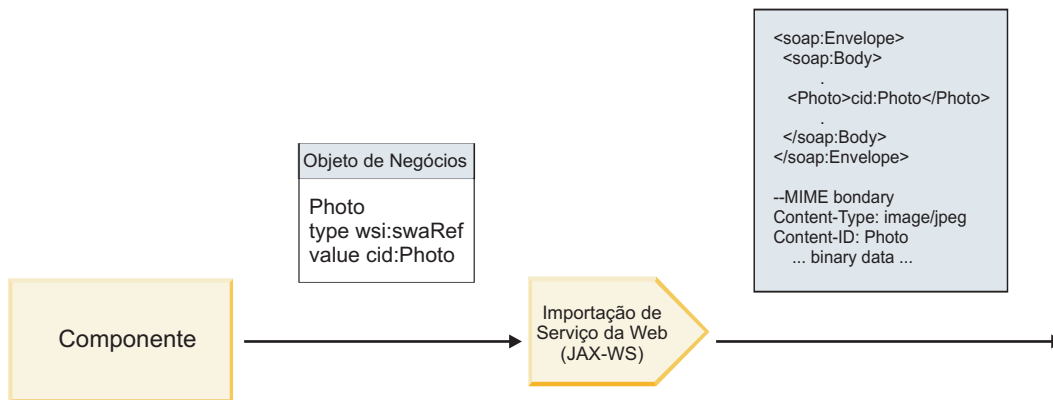
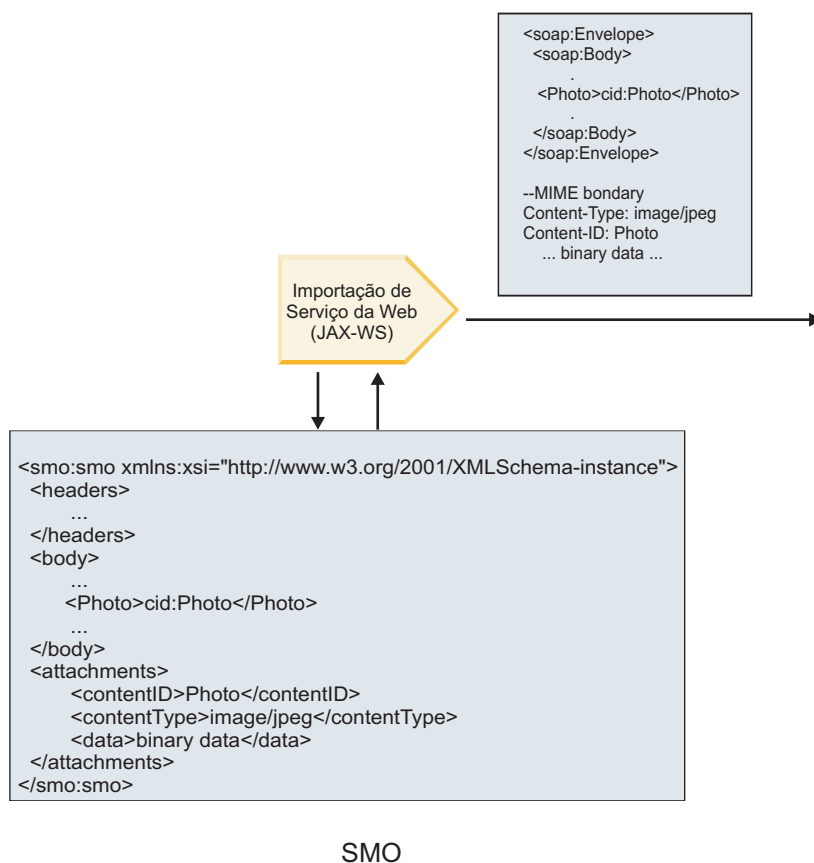


Figura 62. Como a Ligação de Importação (JAX-WS) de Serviço da Web Gera uma Mensagem SOAP com um Anexo swaRef

Configurando Metadados de Anexo em um Componente do Fluxo de Mediação

Se, no SMO, existir um valor de elemento com tipo swaRef e um elemento **attachments**, a ligação preparará a mensagem SOAP (com o anexo) e a enviará para um destinatário.



SMO

Figura 63. Como um Anexo swaRef no SMO é Acessado para Criar a Mensagem SOAP

O elemento **attachments** está presente no SMO apenas se um componente do fluxo de mediação é conectado diretamente à importação ou exportação; ele não é transmitido entre outros tipos de componentes. Se os valores forem necessários em um módulo que contém outros tipos de componentes, um componente do fluxo de mediação deve ser usado para copiar os valores em um local onde eles

podem ser acessados no módulo e outro componente do fluxo de mediação usado para configurar os valores corretos antes de uma chamada de saída por meio de uma importação de serviço da web.

Importante: Como descrito em “Representação XML do SMO,”, a primitiva de mediação Mapeamento transforma mensagens usando uma transformação XSLT 1.0. A transformação opera em uma serialização de XML do SMO. A primitiva de mediação Mapeamento permite que a raiz da serialização seja especificada e o elemento-raiz do documento XML reflita esta raiz.

Quando estiver enviando mensagens SOAP com anexos, o elemento-raiz que você escolheu determinará como os anexos serão propagados.

- Se você usar “/body” como a raiz do mapa XML, todos os anexos serão propagados no mapa por padrão.
- Se você usar “/” como a raiz do mapa, poderá controlar a propagação de anexos.

Anexos referenciados: partes da mensagem em nível superior:

É possível enviar e receber mensagens SOAP que incluem anexos binários que são declarados como partes em sua interface de serviço.

Em uma mensagem SOAP MIME com múltiplas partes, o corpo SOAP é a primeira parte da mensagem e o anexo ou os anexos estão em partes subsequentes.

Qual é a vantagem de enviar ou receber um anexo referenciado em uma mensagem SOAP? Os dados binários que compõem o anexo (que é geralmente muito grande) são mantidos separadamente do corpo da mensagem SOAP para que não precisem ser analisados como XML. Isso resulta em processamento mais eficiente do que se os dados binários fossem mantidos dentro de um elemento XML.

Tipos de Mensagens SOAP com Anexos Referenciados

Começando com a Versão 7.0.0.3 do IBM Business Process Manager, você tem a opção sobre como a mensagem SOAP é gerada:

- **Mensagens Compatíveis com WS-I**

O tempo de execução pode gerar mensagens SOAP que estão em conformidade com os WS-I *Attachments Profile Versão 1.0* e o *WS-I Basic Profile Versão 1.1*. Em uma mensagem SOAP que é compatível com esses perfis, apenas uma parte da mensagem é ligada ao corpo SOAP; para aquelas que são ligadas como anexos, a codificação da parte content-id (conforme descrito no *WS-I Attachments Profile Versão 1.0*) é usada para relacionar o anexo à parte da mensagem.

- **Mensagens Não Compatíveis com WS-I**

O tempo de execução pode gerar mensagens SOAP que não estão em conformidade com os perfis WS-I, mas que são compatíveis com as mensagens geradas na Versão 7.0 ou 7.0.0.2 do IBM Business Process Manager. As mensagens SOAP usam elementos de nível superior denominados após a parte da mensagem com um atributo **href** que contém o anexo **content-id**, mas a codificação da parte content-id (conforme descrito no *WS-I Attachments Profile Versão 1.0*) não é usada.

Selecionando a conformidade de WS-I para exportações de serviço da web

Use o Integration Designer para configurar uma ligação de exportação. Você cria um módulo e sua interface e operações associadas. Você cria uma ligação de serviço da web (JAX-WS). A página Anexos Referenciados exibe todas as partes binárias da operação criada e você pode selecionar quais partes serão anexos. Em seguida, especifique, na página Especifique a Conformidade WS-I AP 1.0 do Integration Designer, uma das seguintes opções:

- **Usar mensagem SOAP compatível com WS-I AP 1.0**

Se você selecionar esta opção, também especificará qual parte da mensagem deve ser ligada ao corpo SOAP.

Nota: Esta opção pode ser usada apenas quando o arquivo WSDL correspondente também é compatível com WS-I.

Um arquivo WSDL que é gerado pelo Integration Designer Versão 7.0.0.3 é compatível com WS-I. No entanto, se você importar um arquivo WSDL que não é compatível com WS-I, não poderá selecionar esta opção.

- **Usar mensagem SOAP não compatível com WS-I AP 1.0**

Se você selecionar esta opção, que é o padrão, a primeira parte da mensagem será ligada ao corpo SOAP.

Nota: Apenas as partes da mensagem de nível superior (ou seja, elementos definidos no portType WSDL como partes dentro da mensagem de entrada ou saída) que possuem um tipo binário (base64Binary ou hexBinary) podem ser enviadas ou recebidas como anexos referenciados.

Consulte o tópico “Trabalhando com Anexos” no centro de informações do Integration Designer para obter informações mais detalhadas.

Para mensagens compatíveis com WS-I, o content-ID que é gerado na mensagem SOAP é uma concatenação dos seguintes elementos:

- O valor do atributo **name** do elemento **wsdl:part** referenciado pelo **mime:content**
- O caractere =
- Um valor globalmente exclusivo, como um UUID
- O caractere @
- Um nome de domínio válido

Processamento de Entrada de Anexos Referenciados

Quando um cliente passa uma mensagem SOAP com um anexo a um componente Service Component Architecture (SCA), a ligação de exportação (JAX-WS) de serviço da web primeiro remove o anexo. Ela, então, analisa a parte SOAP da mensagem e cria um objeto de negócios. Finalmente, a ligação configura o binário de anexo no objeto de negócios.

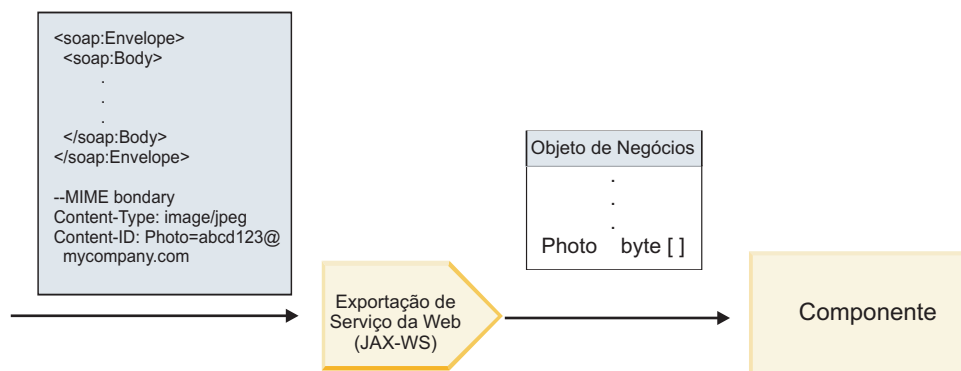


Figura 64. Como a Ligação de Exportação (JAX-WS) de Serviço da Web Processa uma Mensagem SOAP Compatível com WS-I com um Anexo Referenciado

Acessando Metadados de Anexo em um Componente do Fluxo de Mediação

Conforme mostrado em Figura 19 na página 91, quando anexos referenciados são acessados por componentes, os dados do anexo aparecem como uma matriz de bytes.

Cada anexo referenciado de uma mensagem SOAP também possui um elemento **attachments** correspondente no SMO. O elemento **attachments** inclui o tipo de conteúdo de anexo e o caminho para o

elemento de corpo da mensagem quando o anexo é suspenso.

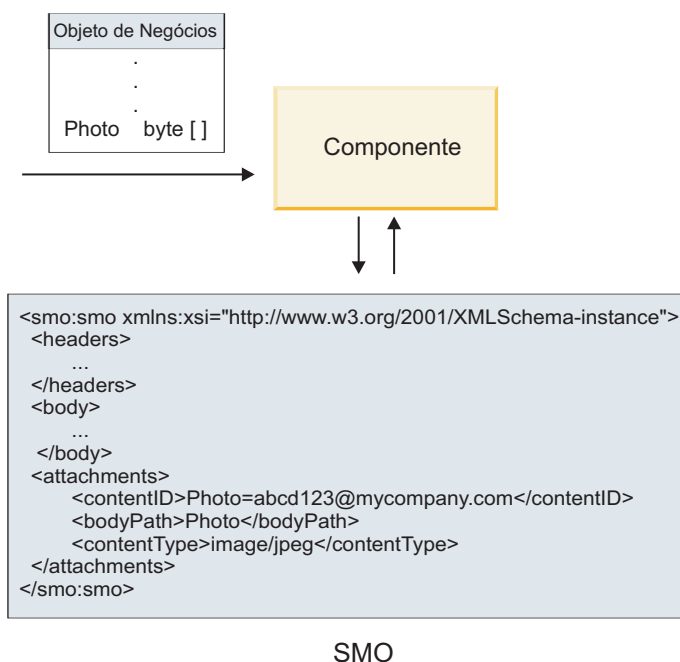


Figura 65. Como Anexos Referenciados Aparecem no SMO

Importante: O caminho para o elemento do corpo da mensagem não é atualizado automaticamente se a mensagem é transformada e o anexo movido. É possível usar um fluxo de mediação para atualizar o elemento **attachments** com o novo caminho (por exemplo, como parte da transformação ou usando um configurador do elemento de mensagem separado).

Como Mensagens SOAP de Saída são Construídas

Você usa o Integration Designer para configurar uma ligação de importação (JAX-WS) de serviço da web para chamar um serviço da web externo. A ligação de importação é configurada com um documento WSDL que descreve o serviço da web a ser chamado e define quais partes da mensagem devem ser passadas como anexos. Também é possível indicar, na página Especifique a Conformidade de WS-I AP 1.0 do Integration Designer, uma das opções a seguir:

- **Usar mensagem SOAP compatível com WS-I AP 1.0**

Se você selecionar esta opção, também especificará qual parte da mensagem deve ser ligada ao corpo SOAP; todas as outras são ligadas a anexos ou cabeçalhos. As mensagens enviadas pela ligação não incluem elementos no corpo SOAP que se referem aos anexos; o relacionamento é expresso por meio do ID de conteúdo do anexo incluindo o nome da parte da mensagem.

- **Usar mensagem SOAP não compatível com WS-I AP 1.0**

Se selecionar esta opção, que é o padrão, a primeira parte da mensagem será ligada ao corpo SOAP; todas as outras serão ligadas aos anexos ou cabeçalhos. As mensagens enviadas pela ligação incluem um ou mais elementos no corpo SOAP que se referem aos anexos por meio de um atributo **href**.

Nota: A parte que representa um anexo, conforme definido no WSDL, deve ser um tipo simples (base64Binary ou hexBinary). Se uma parte for definida por um complexType, essa parte não poderá ser ligada como um anexo.

Processamento de Saída de Anexos Referenciados

A ligação de importação usa informações no SMO para determinar como as partes da mensagem de nível superior binárias são enviadas como anexos.

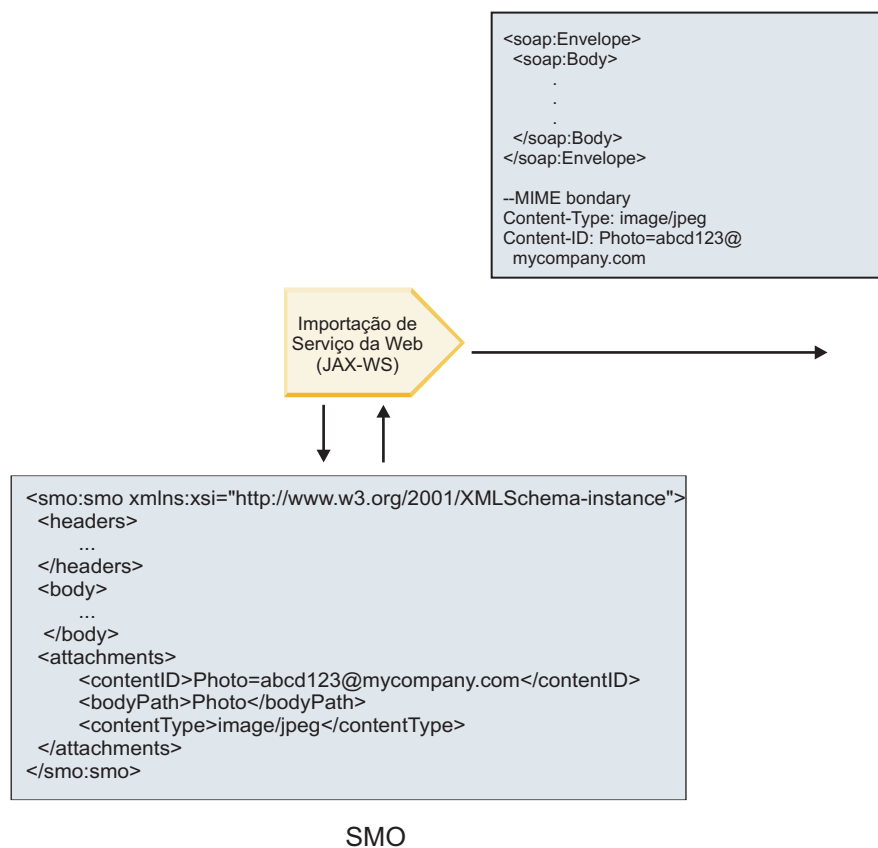


Figura 66. Como o Anexo Referenciado no SMO é Acessado para Criar a Mensagem SOAP

O elemento **attachments** está presente no SMO apenas se um componente do fluxo de mediação é conectado diretamente à importação ou exportação; ele não é transmitido entre outros tipos de componentes. Se os valores forem necessários em um módulo que contém outros tipos de componentes, um componente do fluxo de mediação deve ser usado para copiar os valores em um local onde eles podem ser acessados no módulo e outro componente do fluxo de mediação usado para configurar os valores corretos antes de uma chamada de saída por meio de uma importação de serviço da web.

A ligação usa uma combinação das seguintes condições para determinar como (ou se) a mensagem é enviada:

- Se existe uma ligação MIME WSDL para a parte da mensagem binária de nível superior e, se sim, como o tipo de conteúdo é definido
- Se existe um elemento **attachments** no SMO cujo valor de **bodyPath** referencia uma parte binária de nível superior

Como Anexos são Criados Quando um Elemento attachment Existe no SMO

A tabela a seguir mostra como um anexo é criado e enviado se o SMO contém um elemento **attachment** com um **bodyPath** que corresponde a uma parte do nome da mensagem:

Tabela 61. Como o Anexo é Gerado

Status de ligação MIME WSDL para parte da mensagem binária de nível superior	Como a mensagem é criada e enviada
Presente com um dos seguintes: <ul style="list-style-type: none"> • Nenhum tipo de conteúdo definido para a parte da mensagem • Diversos tipos de conteúdo definidos • Tipo de conteúdo curinga definido 	Parte da mensagem é enviada como um anexo. Content-Id é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, um é gerado. Content-Type é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, é configurado como application/octet-stream .
Presente com conteúdo não curinga único para a parte da mensagem	Parte da mensagem é enviada como um anexo. Content-Id é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, um é gerado. Content-Type é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, é configurado com o tipo definido no elemento de conteúdo MIME WSDL.
Não presente	Parte da mensagem é enviada como um anexo. Content-Id é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, um é gerado. Content-Type é configurado com o valor no elemento <code>attachments</code> , se presente; caso contrário, é configurado como application/octet-stream . Nota: Enviar partes da mensagem como anexos quando não definido como tal no WSDL pode quebrar a conformidade com o WS-I Attachments Profile 1.0 e, portanto, deve ser evitado, se possível.

Como Anexos são Criados Quando Nenhum Elemento `attachment` Existe no SMO

A tabela a seguir mostra como um anexo é criado e enviado se o SMO não contém um elemento `attachment` com um `bodyPath` que corresponde a uma parte do nome da mensagem:

Tabela 62. Como o Anexo é Gerado

Status de ligação MIME WSDL para parte da mensagem binária de nível superior	Como a mensagem é criada e enviada
Presente com um dos seguintes: <ul style="list-style-type: none"> • Nenhum tipo de conteúdo definido para a parte da mensagem • Diversos tipos de conteúdo definidos • Tipo de conteúdo curinga definido 	Parte da mensagem é enviada como um anexo. Content-Id é gerado. Content-Type é configurado como application/octet-stream .
Presente com conteúdo não curinga único para a parte da mensagem	Parte da mensagem é enviada como um anexo. Content-Id é gerado. Content-Type é configurado com o tipo definido no elemento de conteúdo WSDL MIME.
Não presente	Parte da mensagem não é enviada como um anexo.

Importante: Como descrito em “Representação XML do SMO,”, a primitiva de mediação Mapeamento transforma mensagens usando uma transformação XSLT 1.0. A transformação opera em uma serialização

de XML do SMO. A primitiva de mediação Mapeamento permite que a raiz da serialização seja especificada e o elemento-raiz do documento XML reflita esta raiz.

Quando estiver enviando mensagens SOAP com anexos, o elemento-raiz que você escolheu determinará como os anexos serão propagados.

- Se você usar “/body” como a raiz do mapa XML, todos os anexos serão propagados no mapa por padrão.
- Se você usar “/” como a raiz do mapa, poderá controlar a propagação de anexos.

Anexos não referenciados:

Você pode enviar e receber anexos *não referenciados* que não são declarados na interface de serviço.

Em uma mensagem SOAP MIME com várias partes, o corpo SOAP é a primeira parte da mensagem e os anexos estão em partes subsequentes. Nenhuma referência ao anexo é incluída no corpo SOAP.

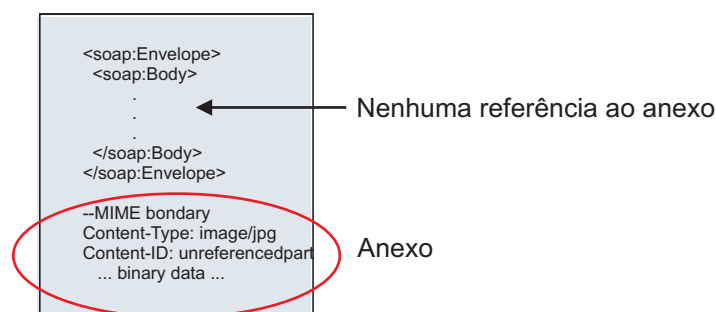


Figura 67. Uma Mensagem SOAP com um Anexo Não Referenciado

É possível enviar uma mensagem SOAP com um anexo não referenciado por meio de uma exportação de serviço da web para uma importação de serviço da web. A mensagem de saída, que é enviada para o serviço da web de destino, contém o anexo.

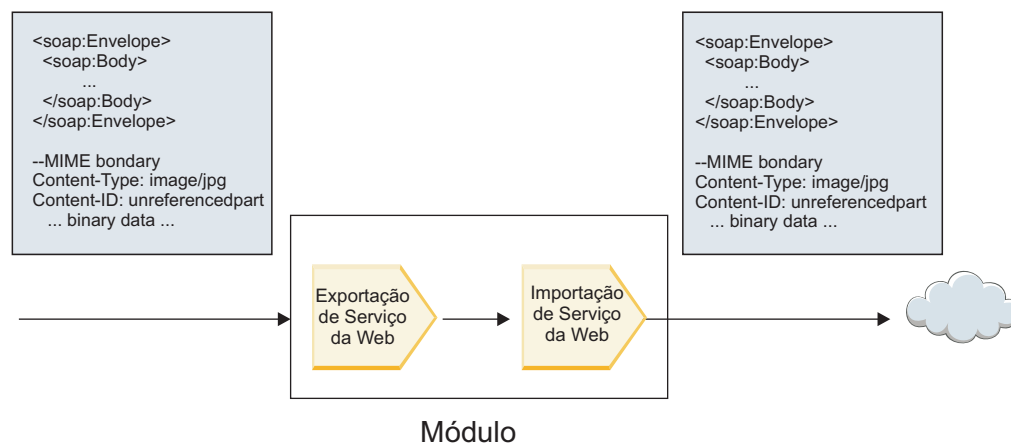


Figura 68. Um Anexo que Passa por meio de um Módulo SCA

No Figura 23 na página 95, a mensagem SOAP, com o anexo, é transmitida sem modificação.

Também é possível modificar a mensagem SOAP usando um componente de fluxo de mediação. Por exemplo, você pode usar o componente do fluxo de mediação para extrair dados da mensagem SOAP (dados binários no corpo da mensagem, nesse caso) e criar uma mensagem SOAP com anexos. Os dados

são processados como parte do elemento anexos de um objeto de mensagem de serviço (SMO).

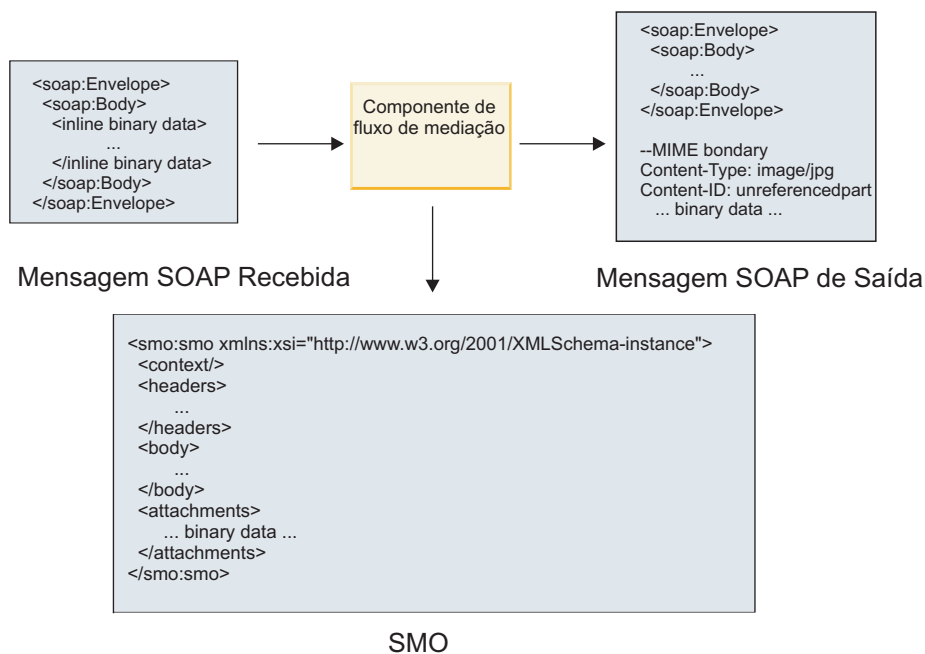


Figura 69. Uma Mensagem Processada por um Componente do Fluxo de Mediação

Em contrapartida, o componente do fluxo de mediação pode transformar a mensagem recebida extraindo e codificando o anexo e, em seguida, transmitindo a mensagem sem anexos.

Em vez de extrair dados de uma mensagem SOAP recebida para formar uma mensagem SOAP com anexos, você pode obter os dados de anexo de uma origem externa, como um banco de dados.

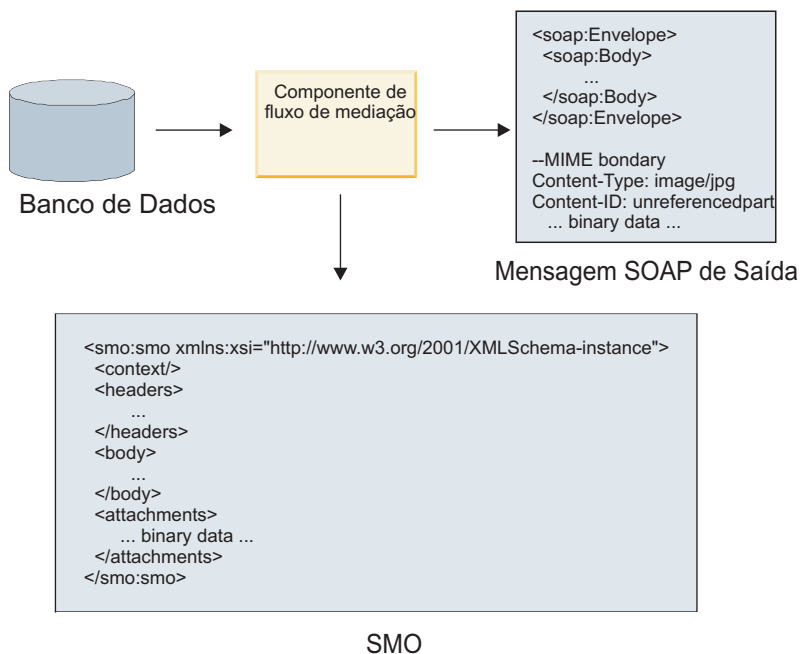


Figura 70. Um Anexo Obtido de um Banco de Dados e Incluído na Mensagem SOAP

De modo inverso, o componente do fluxo de mediação pode extrair o anexo de uma mensagem SOAP recebida e processar a mensagem (por exemplo, armazenar o anexo em um banco de dados).

Anexos não referenciados podem ser propagados apenas entre os componentes do fluxo de mediação. Se um anexo precisar ser acessado ou propagado para outro tipo de componente, use um componente do fluxo de mediação para mover o anexo para um local que seja acessível para esse componente.

Importante: Como descrito em “Representação XML do SMO,”, a primitiva de mediação Mapeamento transforma mensagens usando uma transformação XSLT 1.0. A transformação opera em uma serialização de XML do SMO. A primitiva de mediação Mapeamento permite que a raiz da serialização seja especificada e o elemento-raiz do documento XML reflita esta raiz.

Quando estiver enviando mensagens SOAP com anexos, o elemento-raiz que você escolheu determinará como os anexos serão propagados.

- Se você usar “/body” como a raiz do mapa XML, todos os anexos serão propagados no mapa por padrão.
- Se você usar “/” como a raiz do mapa, poderá controlar a propagação de anexos.

Uso da Ligação de Estilo do Documento WSDL com Mensagens Multipartes:

A organização Web Services Interoperability Organization (WS-I) definiu um conjunto de regras em relação a como os serviços da web devem ser descritos por meio de um WSDL e como as mensagens SOAP correspondentes devem ser formadas a fim de assegurar a interoperabilidade.

Essas regras são especificadas no *WS-I Basic Profile Versão 1.1* (<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>). Em particular, o WS-I Basic Profile 1.1 R2712 afirma: "Uma ligação literal do documento DEVE ser serializada como um ENVELOPE com um soap:Body cujo elemento filho é uma instância da declaração de elemento global referenciada pela parte wsdl:message correspondente".

Isto significa que, ao usar uma ligação SOAP de estilo de documento para uma operação com mensagens (entrada, saída ou falha) que são definidas com diversas partes, apenas uma dessas partes deve ser ligada ao corpo SOAP para ser compatível com o WS-I Basic Profile 1.1.

Além disso, o WS-I Attachments Profile 1.0 R2941 afirma: "Uma wsdl:binding em uma DESCRIÇÃO DEVE ligar cada wsdl:part de uma wsdl:message no wsdl:portType ao qual ele se refere a um soapbind:body, soapbind:header, soapbind:fault, soapbind:headerfault ou mime:content".

Isso significa que, ao usar uma ligação SOAP de estilo de documento para uma operação com mensagens (entrada, saída ou falha) que são definidas com diversas partes, todas as partes diferentes da selecionado a serem ligadas ao corpo SOAP devem ser ligadas como anexos ou cabeçalhos.

A seguinte abordagem é usada quando as descrições WSDL são geradas para exportações com ligações de serviço da web (JAX-WS e JAX-RPC) neste caso:

- É possível escolher qual parte da mensagem é ligada ao corpo SOAP se houver mais de um elemento de tipo não binário. Se houver um único elemento de tipo não binário, esse elemento será automaticamente ligado ao corpo SOAP.
- Para a ligação de JAX-WS, todas as outras partes da mensagem do tipo "hexBinary" ou "Base64Binary" serão ligadas como anexos referenciados. Consulte o “Anexos referenciados: partes da mensagem em nível superior” na página 90.
- Todas as outras partes da mensagem são ligadas como cabeçalhos SOAP.

As ligações de importação JAX-RPC e JAX-WS honram a ligação SOAP em um documento WSDL existente com mensagens de estilo de documento de diversas partes mesmo se não ligar diversas partes ao corpo do SOAP; no entanto, não é possível gerar clientes de serviço da web para documentos WSDL no Rational Application Developer.

Nota: A ligação JAX-RPC não suporta anexos.

O padrão recomendado ao usar mensagens multipartes com uma operação que possui ligação SOAP de estilo de documento é, portanto:

1. Usar o estilo agrupado documento/literal. Nesse caso, as mensagens sempre têm uma única parte; no entanto, os anexos devem ser não referenciados (conforme descrito em “Anexos não referenciados” na página 95) ou com tipo swaRef (conforme descrito em “Anexos referenciados: elementos swaRef-typed” na página 86) neste caso.
2. Usar o estilo RPC/literal. Nesse caso, não há restrições na ligação de WSDL em termos de número de partes ligadas ao corpo SOAP; a mensagem SOAP que resulta sempre possui um único filho que representa a operação que está sendo chamada, com as partes da mensagem sendo filhas desse elemento.
3. Para a ligação de JAX-WS, tenha no máximo uma parte da mensagem que não seja do tipo "hexBinary" ou "Base64Binary", a menos que seja aceitável ligar as outras partes não binárias aos cabeçalhos SOAP.
4. Quaisquer outros casos estão sujeitos ao comportamento descrito.

Nota: Restrições adicionais existem quando você está usando mensagens SOAP que não estão em conformidade com o *WS-I Basic Profile Versão 1.1*.

- A primeira parte da mensagem deve ser não binária.
- Ao receber mensagens SOAP de estilo de documento multipartes com anexos referenciados, a ligação JAX-WS espera que cada anexo referenciado seja representado por um elemento filho do corpo SOAP com um valor de atributo href que identifica o anexo pelo seu ID de conteúdo. A ligação JAX-WS envia anexos referenciados para tais mensagens da mesma maneira. Este comportamento não é compatível com o *WS-I Basic Profile*.

Para assegurar que suas mensagens estejam em conformidade com o *Basic Profile*, siga a abordagem 1 na página 98 ou 2 na página 98 na lista anterior ou evite o uso de anexos referenciados para tais mensagens e use anexos não referenciados ou com tipo swaRef em substituição.

Ligações HTTP

A ligação HTTP é projetada para fornecer conectividade Service Component Architecture (SCA) para HTTP. Consequentemente, aplicativos HTTP existentes ou recém-desenvolvidos podem participar dos ambientes Service Oriented Architecture (SOA).

Hypertext Transfer Protocol (HTTP) é um protocolo amplamente usado para transferir informações na Web. Quando você está trabalhando com um aplicativo externo que usa o protocolo HTTP, uma ligação HTTP é necessária. A ligação HTTP transforma os dados que são passados como uma mensagem no formato nativo para um objeto de negócios em um aplicativo SCA. A ligação HTTP também pode transformar os dados que são atribuídos como um objeto de negócios no formato nativo esperado pelo aplicativo externo.

Nota: Se desejar interagir com clientes e serviços que usam o protocolo SOAP/HTTP de serviços da web, considere usar uma das ligações de serviço da web, que fornecem funcionalidade adicional a respeito da manipulação de qualidades de serviço padrão de serviços da web.

Alguns cenários comuns para usar a ligação HTTP são descritos na lista a seguir:

- Os serviços hospedados pelo SCA podem chamar aplicativos HTTP usando uma importação HTTP.
- Os serviços hospedados pelo SCA podem se expor como aplicativos ativados por HTTP, para que eles possam ser usados pelos clientes HTTP, usando uma exportação HTTP.
- O IBM Business Process Manager e Process Server podem se comunicar entre si por meio de uma infraestrutura de HTTP, consequentemente, os usuários podem gerenciar suas comunicações de acordo com padrões corporativos.

- OIBM Business Process Manager e Process Server podem agir como mediadores de comunicações HTTP, transformação e roteamento de mensagens, o que melhora a integração de aplicativos que usam uma rede HTTP.
- IBM Business Process Manager e Process Server podem ser usados para criar uma ponte entre HTTP e outros protocolos, tais como serviços da Web SOAP/HTTP, adaptadores de recursos baseados em Java Connector Architecture (JCA), JMS e assim por diante.

Informações detalhadas sobre como criar ligações de importação e exportação HTTP podem ser localizadas no centro de informações do Integration Designer. Consulte os tópicos **Desenvolvendo Aplicativos de Integração > Acessando Serviços Externos com HTTP**.

Visão Geral das Ligações HTTP:

A ligação HTTP fornece conectividade para aplicativos hospedados em HTTP. Ela media a comunicação entre aplicativos HTTP e permite que os aplicativos baseados em HTTP existentes sejam chamados a partir de um módulo.

Ligações de Importação HTTP

A ligação de importação HTTP fornece conectividade de saída a partir dos aplicativos Service Component Architecture (SCA) para um servidor HTTP ou aplicativos.

A importação chama uma URL de terminal HTTP. A URL pode ser especificada de uma de três maneiras:

- A URL pode ser configurada dinamicamente nos cabeçalhos HTTP por meio da URL de substituição dinâmica.
- A URL pode ser configurada dinamicamente no elemento de endereço de destino de SMO.
- A URL pode ser especificada como uma propriedade de configuração na importação.

Essa chamada é sempre síncrona por natureza.

Embora chamadas HTTP sempre sejam de solicitação/resposta, a importação de HTTP suporta operações unidirecionais e bidirecionais e ignora a resposta no caso de uma operação unidirecional.

Ligações de Exportação HTTP

A ligação de exportação HTTP fornece conectividade de entrada a partir de aplicativos HTTP para um aplicativo SCA.

Uma URL é definida na exportação HTTP. Aplicativos HTTP que desejam enviar mensagens de solicitação para a exportação usam esta URL para chamar a exportação.

A exportação de HTTP também suporta pings.

Ligações HTTP no Tempo de Execução

Uma importação com uma ligação HTTP no tempo de execução envia uma solicitação com ou sem dados no corpo da mensagem a partir do aplicativo SCA para o serviço da web externo. A solicitação é feita a partir do aplicativo SCA para o serviço da web externo, conforme mostrado no Figura 26 na página 100.

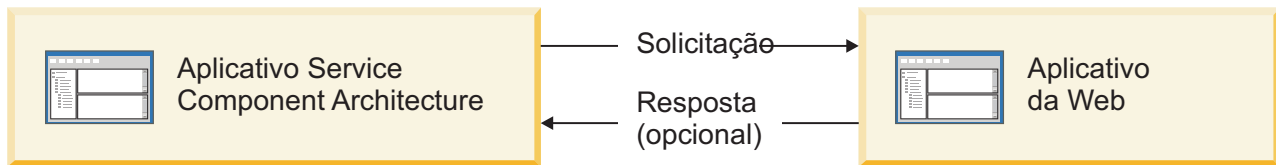


Figura 71. Fluxo de uma Solicitação do Aplicativo SCA para o Aplicativo da Web

Opcionalmente, a importação com a ligação HTTP pode receber dados de volta do aplicativo da web em uma resposta para a solicitação.

Com uma exportação, a solicitação é feita por um aplicativo cliente para um serviço da web, conforme mostrado em Figura 27 na página 100.

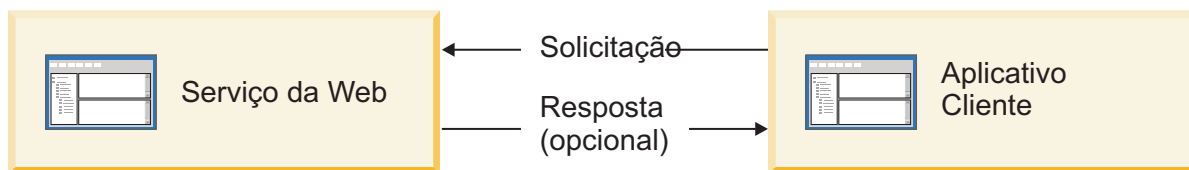


Figura 72. Fluxo de uma solicitação do aplicativo cliente para o serviço da web.

O serviço da Web é um aplicativo da web em execução no servidor. A exportação é implementada nesse aplicativo da Web como um servlet para que o cliente envie sua solicitação para um endereço de URL. O servlet transmite a solicitação para o aplicativo SCA no tempo de execução.

Opcionalmente, a exportação pode enviar dados para o aplicativo cliente em resposta à solicitação.

Cabeçalhos HTTP:

Ligações de importação e exportação HTTP permitem que a configuração de cabeçalhos HTTP e seus valores sejam usados para mensagens de saída. A importação de HTTP usa estes cabeçalhos para solicitações e a exportação de HTTP os usa para respostas.

Os cabeçalhos configurados estaticamente e as informações de controle têm prioridade sobre os valores configurados dinamicamente no tempo de execução. No entanto, os valores de controle da URL, da Versão e do Método de substituição dinâmica substituirão os valores estáticos, que são, de outra forma, considerados os padrões.

A ligação suporta a natureza dinâmica da URL de importação HTTP determinando o valor da URL de destino HTTP, a Versão e o Método no tempo de execução. Estes valores são determinados extraindo o valor de Referência de Terminal, URL de Substituição Dinâmica, Versão e Método.

- Para Referência de Terminal, use as APIs com.ibm.websphere.sca.addressing.EndpointReference ou configure o campo /headers/SMOHeader/Target/address no cabeçalho SMO.
- Para URL de Substituição Dinâmica, Versão e Método, use a seção de parâmetros de controle HTTP da mensagem de Service Component Architecture (SCA). Observe que a URL de Substituição Dinâmica tem precedência sobre a Referência de Terminal de destino; no entanto, a Referência de Terminal se aplica às ligações, por isso, é a abordagem preferencial e deve ser usada quando possível.

As informações de controle e do cabeçalho para mensagens de saída nas ligações de exportação e importação de HTTP são processadas na seguinte ordem:

1. As informações de cabeçalho e controle que excluem a URL de substituição dinâmica HTTP, a Versão e o Método da Mensagem de SCA (prioridade mais baixa)
2. As alterações do console administrativo no nível de exportação/importação

3. As alterações do console administrativo no nível do método da exportação ou importação
4. Endereço de destino especificado pelo caminho da Referência de Terminal ou do cabeçalho SMO
5. URL de Substituição Dinâmica, Versão e Método da mensagem SCA
6. Informações de cabeçalhos e controle do manipulador de dados ou da ligação de dados (prioridade mais alta)

A exportação e importação HTTP irá preencher cabeçalhos de direção de entrada e parâmetros de controle com dados da mensagem de entrada (HTTPExportRequest e HTTPImportResponse) somente se a propagação de cabeçalho de protocolo for configurada como **True**. Inversamente, a exportação e importação HTTP irão ler e processar cabeçalhos de saída e parâmetros de controle (HTTPExportResponse e HTTPImportRequest) somente se a propagação de cabeçalho de protocolo for configurada como **True**.

Nota: As alterações de ligação de dados ou do manipulador de dados para cabeçalhos ou parâmetros de controle na resposta de importação ou solicitação de exportação não alterarão as instruções de processamento da mensagem dentro da ligação de importação ou exportação e devem ser usadas apenas para propagar valores modificados para componentes SCA de recebimento de dados.

O serviço de contexto é responsável por propagar o contexto (incluindo os cabeçalhos de protocolo, tal como o cabeçalho HTTP e o contexto do usuário, tal como o ID da conta) juntamente com um caminho de chamada SCA. Durante o desenvolvimento no IBM Integration Designer, é possível controlar a propagação de contexto por meio de propriedades de importação e exportação. Para obter detalhes adicionais, consulte as informações de ligações de importação e exportação no centro de informações do IBM Integration Designer.

Estruturas de Cabeçalho HTTP e Suporte Fornecidos

O Tabela 34 na página 101 detalha em itens os parâmetros de solicitação/resposta para solicitações e respostas de Importação HTTP e Exportação HTTP.

Tabela 63. Informações do Cabeçalho HTTP Fornecidas

Nome do Controle	Solicitação de Importação de HTTP	Resposta de Importação de HTTP	Solicitação de Exportação de HTTP	Resposta de Exportação de HTTP
URL	Ignorado	Não Configurado	Ler a partir da mensagem de solicitação. Nota: A sequência de consultas também faz parte do parâmetro de controle da URL.	Ignorado
Versão (valores possíveis: 1.0, 1.1; o padrão é 1.1)	Ignorado	Não Configurado	Ler a partir da mensagem de solicitação	Ignorado
Método	Ignorado	Não Configurado	Ler a partir da mensagem de solicitação	Ignorado

Tabela 63. Informações do Cabeçalho HTTP Fornecidas (continuação)

Nome do Controle	Solicitação de Importação de HTTP	Resposta de Importação de HTTP	Solicitação de Exportação de HTTP	Resposta de Exportação de HTTP
URL de Substituição Dinâmica	Se configurada no manipulador de dados ou na ligação de dados, substitui a URL de Importação HTTP. Gravado na mensagem na linha de solicitação. Nota: A sequência de consultas também faz parte do parâmetro de controle da URL.	Não Configurado	Não Configurado	Ignorado
Versão de Substituição Dinâmica	Se configurada, substitui a Versão de Importação de HTTP. Gravado na mensagem na linha de solicitação.	Não Configurado	Não Configurado	Ignorado
Método de Substituição Dinâmica	Se configurado, substitui o Método de Importação de HTTP. Gravado na mensagem na linha de solicitação.	Não Configurado	Não Configurado	Ignorado
Tipo de Mídia (Este parâmetro de controle transporta parte do valor do cabeçalho HTTP Content-Type.)	Se presente, gravado na mensagem como parte do cabeçalho Content-Type. Nota: Este valor de elemento de controle deve ser fornecido pelo manipulador de dados ou pela ligação de dados.	Ler a partir da mensagem de resposta, cabeçalho Content-Type	Ler a partir da mensagem de solicitação, cabeçalho Content-Type	Se presente, gravado na mensagem como parte do cabeçalho Content-Type. Nota: Este valor de elemento de controle deve ser fornecido pelo manipulador de dados ou pela ligação de dados.
Conjunto de Caracteres (padrão: UTF-8)	Se presente, gravado na mensagem como parte do cabeçalho Content-Type. Nota: Este valor de elemento de controle deve ser fornecido pela ligação de dados.	Ler a partir da mensagem de resposta, cabeçalho Content-Type	Ler a partir da mensagem de solicitação, cabeçalho Content-Type	Suportado; gravado na mensagem como parte do cabeçalho Content-Type. Nota: Este valor de elemento de controle deve ser fornecido pela ligação de dados.
Codificação de Transferência (Valores possíveis : em partes, identidade; o padrão é identidade)	Se presente, gravado na mensagem como um cabeçalho e controla como a transformação da mensagem é codificada.	Leia a partir da mensagem de resposta	Ler a partir da mensagem de solicitação	Se presente, gravado na mensagem como um cabeçalho e controla como a transformação da mensagem é codificada.

Tabela 63. Informações do Cabeçalho HTTP Fornecidas (continuação)

Nome do Controle	Solicitação de Importação de HTTP	Resposta de Importação de HTTP	Solicitação de Exportação de HTTP	Resposta de Exportação de HTTP
Codificação de Conteúdo (Valores possíveis: gzip, x-gzip, deflate, identidade; o padrão é identidade)	Se presente, gravado na mensagem como um cabeçalho e controla como a carga útil é codificada.	Leia a partir da mensagem de resposta	Ler a partir da mensagem de solicitação	Se presente, gravado na mensagem como um cabeçalho e controla como a carga útil é codificada.
Content-Length	Ignorado	Leia a partir da mensagem de resposta	Ler a partir da mensagem de solicitação	Ignorado
StatusCode (padrão: 200)	Não suportado	Leia a partir da mensagem de resposta	Não suportado	Se presente, gravado na mensagem na linha de resposta
ReasonPhrase (padrão: OK)	Não suportado	Leia a partir da mensagem de resposta	Não suportado	Valor de controle ignorado. O valor da linha de resposta da mensagem é gerado a partir do StatusCode.
Autenticação (contém diversas propriedades)	Se presente, usada para construir o cabeçalho de Autenticação Básica. Nota: O valor para este cabeçalho será codificado apenas no protocolo HTTP. No SCA, ele será decodificado e transmitido como texto não criptografado.	Não aplicável	Ler a partir do cabeçalho de Autenticação Básica da mensagem de solicitação. A presença deste cabeçalho não indica que o usuário foi autenticado. A autenticação deve ser controlada na configuração do servlet. Nota: O valor para este cabeçalho será codificado apenas no protocolo HTTP. No SCA, ele será decodificado e transmitido como texto não criptografado.	Não aplicável
Proxy (contém diversas propriedades: Host, Porta, Autenticação)	Se presente, usado para estabelecer a conexão por meio do proxy.	Não aplicável	Não aplicável	Não aplicável
SSL (contém diversas propriedades: Keystore, Senha do Keystore, Trustore, Senha do Trustore, ClientAuth)	Se preenchido e a URL de destino for HTTPS, ele será usado para estabelecer uma conexão por meio de SSL.	Não aplicável	Não aplicável	Não aplicável

Ligações de Dados HTTP:

Para cada mapeamento diferente de dados entre uma mensagem Service Component Architecture (SCA) e uma mensagem de protocolo HTTP, um manipulador de dados ou uma ligação de dados HTTP deve ser configurada. Os manipuladores de dados fornecem uma interface de ligação neutra que permite a reutilização através de ligações de transporte e representam a abordagem recomendada; ligações de dados são específicas para uma ligação de transporte específica. Classes de ligação de dados específicas de HTTP são fornecidas; também é possível gravar manipuladores de dados ou ligações de dados customizadas.

Nota: As três classes de ligação de dados HTTP descritas neste tópico (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML e HTTPServiceGatewayDataBinding) estão descontinuadas a partir do IBM Business Process Manager Versão 7.0. Em vez de usar as ligações de dados descritas neste tópico, considere os manipuladores de dados a seguir:

- Use SOAPDataHandler em vez de HTTPStreamDataBindingSOAP.
- Use UTF8XMLDataHandler em vez de HTTPStreamDataBindingXML
- Use GatewayTextDataHandler em vez de HTTPServiceGatewayDataBinding

As ligações de dados são fornecidas para uso com importações HTTP e exportações HTTP: ligação de dados binários, ligação de dados XML e ligação de dados SOAP. Uma ligação de dados de resposta não é requerida para operações unidirecionais. Uma ligação de dados é representada pelo nome de uma classe Java cujas instâncias podem converter ambos de HTTP para ServiceDataObject e vice-versa. Um seletor de função deve ser usado em uma exportação que, em conjunto com ligações de método, pode determinar qual ligação de método é usada e qual operação é chamada. As ligações de dados fornecidas são:

- Ligações de dados binários, que tratam o corpo como dados binários não estruturados. A implementação do esquema XSD de ligação de dados binários é conforme a seguir:

```
<xsd:schema elementFormDefault="qualified"
  targetNamespace="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:tns="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:complexType name="HTTPBaseBody">
    <xsd:sequence/>
  </xsd:complexType>

  <xsd:complexType name="HTTPBytesBody">
    <xsd:complexContent>
      <xsd:extension base="tns:HTTPBaseBody">
        <xsd:sequence>
          <xsd:element name="value" type="xsd:hexBinary"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

- Ligações de dados XML, que suportam o corpo como dados XML. A implementação da ligação de dados XML é semelhante à ligação de dados XML de JMS e não tem restrições sobre o esquema de interface.
- Ligações de dados SOAP, que suportam o corpo como dados SOAP. A implementação da ligação de dados SOAP não possui restrições sobre o interface de interface.

Implementando Ligações de Dados HTTP Customizadas

Esta seção descreve como implementar uma ligação de dados HTTP customizada.

Nota: A abordagem recomendada é implementar um manipulador de dados customizado porque ele pode ser reutilizado em ligações de transporte.

HTTPStreamDataBinding é a interface principal para manipular mensagens HTTP customizadas. A interface foi projetada para permitir a manipulação de cargas úteis grandes. No entanto, para que essa implementação funcione, essa ligação de dados deve retornar as informações e cabeçalhos de controle antes de gravar a mensagem no fluxo.

Os métodos e suas ordens de execução, listados abaixo, devem ser implementados pela ligação de dados customizada.

Para customizar uma ligação de dados, grave uma classe que implementa HTTPStreamDataBinding. A ligação de dados deve ter quatro propriedades privadas:

- private DataObject pDataObject
- private HTTPControl pCtrl
- private HTTPHeaders pHeaders
- private yourNativeDataType nativeData

A ligação HTTP chamará a ligação de dados customizados na seguinte ordem:

- Processamento de saída (DataObject para o formato Nativo):
 1. setDataObject(...)
 2. setHeaders(...)
 3. setControlParameters(...)
 4. setBusinessException(...)
 5. convertToNativeData()
 6. getControlParameters()
 7. getHeaders()
 8. write(...)
- Processamento de entrada (Formato Nativo para DataObject):
 1. setControlParameters(...)
 2. setHeaders(...)
 3. convertFromNativeData(...)
 4. isBusinessException()
 5. getDataObject()
 6. getControlParameters()
 7. getHeaders()

É necessário chamar setDataObject(...) em convertFromNativeData(...) para configurar o valor de dataObject, o qual é convertido de dado nativo na propriedade privada "pDataObject".

```
public void setDataObject(DataObject dataObject)
    throws DataBindingException {
    pDataObject = dataObject;
}

public void setControlParameters(HTTPControl arg0) {
    this.pCtrl = arg0;
}

public void setHeaders(HTTPHeaders arg0) {
    this.pHeaders = arg0;
}

/*
 * Add http header "IsBusinessException" in pHeaders.
 * Two steps:
 * 1.Remove all the header with name IsBusinessException (case-insensitive) first.
 * This is to make sure only one header is present.
```

```

* 2.Add the new header "IsBusinessException"
*/
public void setBusinessException(boolean isBusinessException) {
    //remove all the header with name IsBusinessException (case-insensitive) first.
    //This is to make sure only one header is present.
    //add the new header "IsBusinessException", code example:
    HTTPHeader header=HeadersFactory.eINSTANCE.createHTTPHeader();
    header.setName("IsBusinessException");
    header.setValue(Boolean.toString(isBusinessException));
    this.pHeaders.getHeader().add(header);
}
public HTTPControl getControlParameters() {
    return pCtrl;
}
public HTTPHeaders getHeaders() {
    return pHeaders;
}
public DataObject getDataObject() throws DataBindingException {
    return pDataObject;
}
/*
* Get header "IsBusinessException" from pHeaders, return its boolean value
*/
public boolean isBusinessException() {
    String headerValue = getHeaderValue(pHeaders,"IsBusinessException");
    boolean result=Boolean.parseBoolean(headerValue);
    return result;
}
public void convertToNativeData() throws DataBindingException {
    DataObject dataObject = getDataObject();
    this.nativeData=realConvertWorkFromSD0ToNativeData(dataObject);
}
public void convertFromNativeData(HTTPInputStream arg0){
    //Customer-developed method to
    //Read data from HTTPInputStream
    //Convert it to DataObject
    DataObject dataobject=realConvertWorkFromNativeDataToSD0(arg0);
    setDataObject(dataobject);
}
public void write(HTTPOutputStream output) throws IOException {
    if (nativeData != null)
        output.write(nativeData);
}

```

Ligações EJB

Ligações de importação de Enterprise JavaBeans (EJB) permitem que os componentes de Service Component Architecture (SCA) chamem serviços fornecidos pela lógica de negócios Java EE em execução em um servidor Java EE. Ligações de exportação EJB permitem que componentes SCA sejam expostos como Enterprise JavaBeans de forma que a lógica de negócios Java EE possa chamar componentes SCA de outra forma indisponíveis para elas.

Ligações de importação EJB:

Ligações de importação EJB permitem que um módulo SCA chame as implementações EJB especificando a forma com que o módulo de consumo é ligado ao EJB externo. A importação de serviços de uma implementação EJB externa permite que os usuários conectem sua lógica de negócios no ambiente do IBM Business Process Manager e participem de um processo de negócios.

Você pode usar o Integration Designer para criar ligações de importação de EJB. É possível usar um dos seguintes procedimentos para gerar as ligações:

- Criando a importação de EJB usando o assistente de serviço externo

É possível usar o assistente de serviço externo no Integration Designer para construir uma importação de EJB baseada em uma implementação existente. O assistente de serviço externo cria serviços com base nos critérios que você fornece. Ele, então, gera objetos de negócios, interfaces e arquivos de importação com base nos serviços descobertos.

- Criando a importação de EJB usando o editor de montagem

É possível criar uma importação de EJB dentro de um diagrama de conjunto usando o editor de montagem do Integration Designer. Na paleta, é possível usar uma Importação ou usar uma classe Java para criar a ligação EJB.

A importação gerada possui ligações de dados que fazem a conexão Java-WSDL em vez de requerer um componente de ponte Java. É possível ligar diretamente um componente a uma referência de Web Services Description Language (WSDL) na importação de EJB que se comunica com um serviço baseado em EJB usando uma interface Java.

A importação de EJB pode interagir com a lógica de negócios Java EE usando o modelo de programação EJB 2.1 ou o modelo de programação EJB 3.0.

A chamada para a lógica de negócios Java EE pode ser local (apenas para EJB 3.0) ou remota.

- A chamada local é usada quando você deseja chamar a lógica de negócios Java EE que reside no mesmo servidor que a importação.

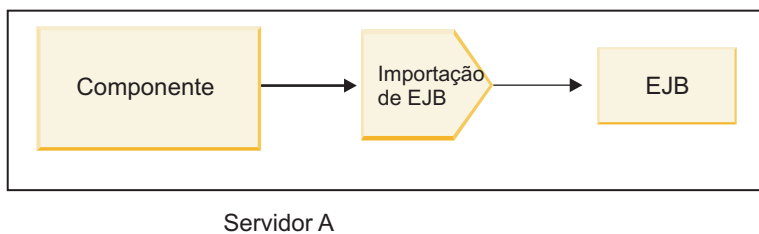


Figura 73. Chamada Local de um EJB (Apenas EJB 3.0)

- A chamada remota é usada quando você deseja chamar a lógica de negócios Java EE que não reside no mesmo servidor que a importação.

Por exemplo, na figura a seguir, uma importação EJB usa o Remote Method Invocation over Internet InterORB Protocol (RMI/IIOP) para chamar um método EJB em outro servidor.

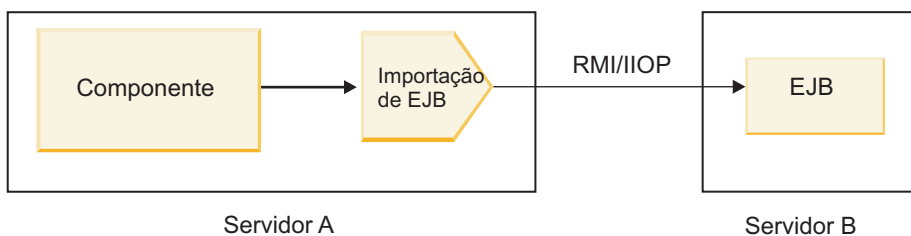


Figura 74. Chamada Remota de um EJB

Quando ele configura a ligação EJB, o Integration Designer usa o nome JNDI para determinar o nível de modelo de programação EJB e o tipo de chamada (local ou remota).

Ligações de importação EJB contêm os seguintes componentes principais:

- Manipulador de dados JAX-WS
- Seletor de falhas EJB
- Seletor de função de importação EJB

Se seu cenário do usuário não for baseado no mapeamento de JAX-WS, você poderá precisar de um manipulador de dados customizado, do seletor de função e do seletor de falha para executar as tarefas de outra forma concluídas pelos componentes que fazem parte das ligações de importação de EJB. Isso inclui o mapeamento normalmente concluído pelo algoritmo de mapeamento customizado.

Ligações de exportação EJB:

Aplicativos Java EE externos podem chamar um componente SCA por meio de uma ligação de exportação EJB. O uso de uma exportação EJB permite que você exponha componentes SCA para que aplicativos Java EE externos possam chamar esses componentes usando o modelo de programação EJB.

Nota: A exportação EJB é um bean stateless.

Use o Integration Designer para criar ligações EJB. É possível usar um dos seguintes procedimentos para gerar as ligações:

- Criando ligações de exportação EJB usando o assistente de serviço externo

É possível usar o assistente de serviço externo no Integration Designer para construir um serviço de exportação de EJB baseado em uma implementação existente. O assistente de serviço externo cria serviços com base nos critérios que você fornece. Em seguida, ele gera objetos de negócios, interfaces e arquivos de exportação com base nos serviços descobertos.

- Criando ligações de exportação EJB usando o editor de montagem

É possível criar uma exportação de EJB usando o editor de montagem do Integration Designer.

Importante: Um cliente Java 2 Platform, Standard Edition (J2SE) não pode chamar o cliente de exportação EJB que é gerado no Integration Designer.

É possível gerar a ligação a partir de um componente SCA existente ou você pode gerar uma exportação com uma ligação EJB para uma interface Java.

- Quando você gera uma exportação para um componente SCA existente que possui uma interface WSDL existente, a exportação tem a uma interface Java designada.
- Quando você gera uma exportação para uma interface Java, é possível selecionar um WSDL ou uma interface Java para a exportação.

Nota: Uma interface Java usada para criar uma exportação de EJB possui as seguintes limitações com relação aos objetos (parâmetros de entrada e saída e exceções) transmitidos como parâmetros em uma chamada remota:

- Eles devem ser do tipo concreto (em vez de uma interface ou tipo abstrato).
- Eles devem estarem conformidade com a especificação do Enterprise JavaBeans. Eles devem ser serializáveis e ter o construtor sem argumento padrão e todas as propriedades devem ser acessíveis por meio dos métodos getter e setter.

Consulte o website da Sun Microsystems, Inc. em <http://java.sun.com> para obter informações sobre a especificação de Enterprise JavaBeans.

Além disso, a exceção deve ser uma exceção verificada, herdada de `java.lang.Exception` e deve ser singular (ou seja, ela não suporta emitir diversos tipos de exceção verificada).

Observe também que a interface de negócios de um Java EnterpriseBean é uma interface Java simples e não deve estender `javax.ejb.EJBObject` ou `javax.ejb.EJBLocalObject`. Os métodos da interface de negócios não deve emitir `java.rmi.remote.exception`.

As ligações de exportação de EJB podem interagir com a lógica de negócios Java EE usando o modelo de programação EJB 2.1 ou o modelo de programação EJB 3.0.

A chamada pode ser local (apenas para EJB 3.0) ou remota.

- A chamada Local é usada quando a lógica de negócios Java EE chama um componente SCA que reside no mesmo servidor que a exportação.
- A chamada remota é usada quando a lógica de negócios Java EE não reside no mesmo servidor que a exportação.

Por exemplo, na figura a seguir, um EJB usa RMI/IIOP para chamar um componente SCA em um servidor diferente.

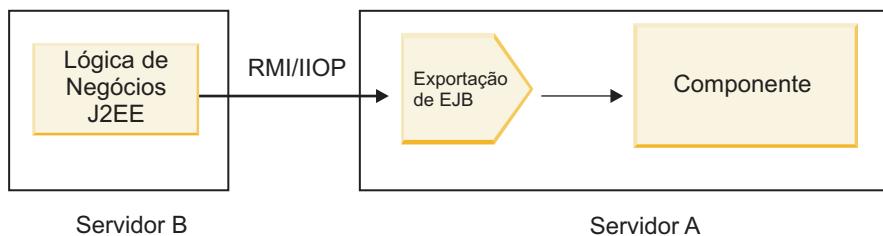


Figura 75. Chamada remota de um cliente para um componente SCA por meio de uma exportação de EJB

Quando ele configura a ligação EJB, o Integration Designer usa o nome JNDI para determinar o nível de modelo de programação EJB e o tipo de chamada (local ou remota).

As ligações de exportação EJB contêm os principais componentes a seguir:

- Manipulador de dados JAX-WS
- Seletor de função de exportação EJB

Se seu cenário do usuário não for baseado no mapeamento de JAX-WS, você poderá precisar de um manipulador de dados customizado e o seletor de função para executar as tarefas de outra forma concluídas pelos componentes que fazem parte das ligações de exportação de EJB. Isso inclui o mapeamento normalmente concluído pelo algoritmo de mapeamento customizado.

Propriedades de ligação EJB:

As ligações de importação EJB usam seus nomes JNDI configurados para determinar o nível de modelo de programação EJB e o tipo de chamada (local ou remoto). As ligações de importação e exportação EJB usam o manipulador de dados JAX-WS para transformação de dados. A ligação de importação EJB usa um seletor de função de importação EJB e um seletor de falhas EJB e a ligação de exportação EJB usa um seletor de função de exportação EJB.

Nomes JNDI e ligações de importação EJB:

Quando ele configura a ligação EJB em uma importação, o Integration Designer usa o nome JNDI para determinar o nível de modelo de programação EJB e o tipo de chamada (local ou remota).

Se nenhum nome JNDI for especificado, a ligação de interface EJB padrão será usada. Os nomes padrão que são criados dependem se você está chamando o EJB 2.1 JavaBeans ou o EJB 3.0 JavaBeans.

Nota: Consulte o tópico "Visão Geral de Ligações do Aplicativo EJB 3.0" no centro de informações do WebSphere Application Server para obter informações detalhadas sobre convenções de nomenclatura.

- JavaBeans EJB 2.1

O nome JNDI padrão pré-selecionado pelo Integration Designer é a ligação do EJB 2.1 padrão, que usa o formato **ejb/** mais a interface inicial, separados por barras.

Por exemplo, para a interface inicial do EJB 2.1 JavaBeans para com.mycompany.myremotebusinesshome, a ligação padrão é:

```
ejb/com/mycompany/myremotebusinesshome
```

Para EJB 2.1, apenas a chamada EJB remota é suportada.

- JavaBeans EJB 3.0

O nome JNDI padrão pré-selecionado pelo Integration Designer para o JNDI local é o nome de classe completo da interface local precedido por **ejblocal:**. Por exemplo, para a interface completa da interface `com.mycompany.mylocalbusiness` local, o JNDI EJB 3.0 pré-selecionado é:

```
ejblocal:com.mycompany.mylocalbusiness
```

Para a interface `com.mycompany.myremotebusiness` remota, o JNDI EJB 3.0 pré-selecionado é a interface completa:

```
com.mycompany.myremotebusiness
```

As ligações de aplicativos padrão EJB 3.0 são descritas no local a seguir: visão geral das ligações de aplicativos EJB 3.0. Visão geral de ligações de aplicativos EJB 3.0.

O Integration Designer usará o nome "abreviado" como o local de JNDI padrão para EJBs que usam o modelo de programação da versão 3.0.

Nota: Se a referência JNDI implementada do EJB de destino for diferente do local da ligação JNDI padrão porque um mapeamento customizado foi usado ou configurado, o nome JNDI de destino deverá ser especificado corretamente. É possível especificar o nome no Integration Designer antes da implementação ou, para a ligação de importação, é possível alterar o nome no console administrativo (após a implementação) para corresponder ao nome JNDI do EJB de destino.

Para obter informações adicionais sobre a criação de ligações EJB, consulte a seção dedicada a Trabalhar com Ligações EJB no centro de informações do Integration Designer.

Manipulador de dados JAX-WS:

A ligação de importação do Enterprise JavaBeans (EJB) usa o manipulador de dados do JAX-WS para transformar os objetos de negócios de solicitação em parâmetros de objeto Java e para transformar o valor de retorno do objeto Java no objeto de negócios de resposta. A ligação de exportação do EJB usa o manipulador de dados JAX-WS para transformar EJBs de solicitação nos objetos de negócios de solicitação e para transformar o objeto de negócios de resposta em um valor de retorno.

Esse manipulador de dados mapeia dados da interface WSDL especificada por SCA para a interface Java EJB de destino (e vice-versa) usando a especificação Java API para XML Web Services (JAX-WS) e a especificação Java Architecture para XML Binding (JAXB).

Nota: O suporte atual é restrito às especificações JAX-WS 2.1.1 e JAXB 2.1.3.

O manipulador de dados especificado no nível de ligação EJB é usado para executar o processamento de solicitação, de resposta, de falha e de exceção de tempo de execução.

Nota: Para falhas, um manipulador de dados específico pode ser especificado para cada falha especificando a propriedade de configuração `faultBindingType`. Isso substitui o valor especificado no nível de ligação EJB.

O manipulador de dados JAX-WS é usado por padrão quando a ligação EJB possui uma interface WSDL. Este manipulador de dados não pode ser usado para transformar uma mensagem SOAP que representa uma chamada JAX-WS em um objeto de dados.

A ligação de importação de EJB usa um manipulador de dados para transformar um objeto de dados em uma matriz de Objeto Java (`Object[]`). Durante as comunicações de saída, ocorre o seguinte processamento:

1. A ligação EJB configura o tipo esperado, o elemento esperado e o nome do método de destino no `BindingContext` para corresponder àqueles especificados no WSDL.
2. A ligação EJB chama o método de transformação para o objeto de dados que requer a transformação de dados.

3. O manipulador de dados retorna um `Object[]` que representa os parâmetros do método (na ordem de suas definições no método).
4. A ligação EJB usa o `Object[]` para chamar o método na interface EJB de destino.

A ligação também prepara um `Object[]` para processar a resposta a partir da chamada EJB.

- O primeiro elemento no `Object[]` é o valor de retorno da chamada do método Java.
- Os valores subsequentes representam os parâmetro de entrada para o método.

Isto é requerido para suportar os tipos Entrada/Saída e Saída de parâmetros.

Para parâmetros de tipo Saída, os valores devem ser retornados no objeto de dados de resposta.

O manipulador de dados processa e transforma valores localizados no `Object[]` e, em seguida, retorna uma resposta ao objeto de dados.

O manipulador de dados suporta `xs:AnyType`, `xs:AnySimpleType` e `xs:Any` juntamente com outros tipos de dados XSD. Para ativar o suporte para `xs:Any`, use o **@XmlAnyElement (lax=true)** para a propriedade JavaBeans no código Java, como mostra o seguinte exemplo:

```
public class TestType {
    private Object[] object;

    @XmlAnyElement (lax=true)
    public Object[] getObject() {
        return object;
    }

    public void setObject (Object[] object) {
        this.object=object;
    }
}
```

Isto torna o objeto de propriedade em `TestType` um campo `xs:any`. O valor de classe Java usado no campo `xs:any` deve ter a anotação **@XmlAnyElement**. Por exemplo, se `Address` for a classe Java que está sendo usada para preencher a matriz do objeto, a classe `Address` deverá ter a anotação **@XmlRootElement**.

Nota: Para customizar o mapeamento a partir do tipo XSD para tipos Java definidos pela especificação JAX-WS, altere as anotações JAXB para se ajustarem às suas necessidades comerciais. O manipulador de dados JAX-WS suporta `xs:any`, `xs:anyType` e `xs:anySimpleType`.

As restrições a seguir são aplicáveis para o manipulador de dados JAX-WS:

- O manipulador de dados não inclui suporte para a anotação **@WebParam** do atributo de cabeçalho.
- O namespace para arquivos de esquema de objeto de negócios (arquivos XSD) não inclui mapeamento padrão a partir do nome do pacote Java. A anotação **@XMLSchema** em `package-info.java` também não funciona. A única maneira de criar um XSD com um namespace é usar as anotações **@XmlType** e **@XmlRootElement**. **@XmlRootElement** define o namespace de destino para o elemento global nos tipos do JavaBeans.
- O assistente de importação EJB não cria arquivos XSD para classes não relacionadas. A Versão 2.0 não suporta a anotação **@XmlSeeAlso**, portanto, se a classe-filha não for referida diretamente a partir da classe-pai, um XSD não será criado. A solução para esse problema é executar `SchemaGen` para essas classes-filhas.

`SchemaGen` é um utilitário da linha de comandos (localizado no diretório `WPS_Install_Home/bin`) fornecido para criar os arquivos XSD para um determinado bean. Esses XSDs devem ser copiados manualmente no módulo para que a solução funcione.

Seletor de Falha EJB:

O seletor de falha EJB determina se uma chamada EJB resultou em uma falha, uma exceção de tempo de execução ou uma resposta bem-sucedida.

Se uma falha for detectada, o seletor de falha EJB retornará o nome da falha nativo para o tempo de execução de ligação para que o manipulador de dados JAX-WS possa converter o objeto de exceção em um objeto de negócios de falha.

Em uma resposta bem-sucedida (não falha), a ligação de importação EJB monta uma matriz de objeto Java (Object[]) para retornar os valores.

- O primeiro elemento no Object[] é o valor de retorno da chamada do método Java.
- Os valores subsequentes representam os parâmetro de entrada para o método.

Isto é requerido para suportar os tipos Entrada/Saída e Saída de parâmetros.

Para cenários de exceção, a ligação monta um Object[] e o primeiro elemento representa a exceção emitida pelo método.

O seletor de falha pode retornar qualquer um dos seguintes valores:

Tabela 64. Valores de Retorno

Tipo	Valor de Retorno	Descrição
Falha	ResponseType.FAULT	Retornado quando o Object[] transmitido contém um objeto de exceção.
Exceção de Tempo de Execução	ResponseType.RUNTIME	Retornado se o objeto de exceção não corresponde a nenhum dos tipos de exceção declarada no método.
Resposta Normal	ResponseType.RESPONSE	Retornado em todos os outros casos.

Se o seletor de falha retorna um valor de **ResponseType.FAULT**, o nome de falha nativo é retornado. Este nome de falha nativo é usado pela ligação para determinar o nome da falha WSDL correspondente a partir do modelo e para chamar o manipulador de dados de falha correto.

Seletor de função EJB:

As ligações EJB usam um seletor de função de importação (para processamento de saída) ou um seletor de função de exportação (para processamento de entrada) para determinar o método EJB a chamar.

Seletor de Função de Importação

Para processamento de saída, o seletor de função de importação deriva o tipo de método EJB com base no nome da operação chamada pelo componente SCA que é ligada à importação EJB. O seletor de função procurará a anotação @WebMethod na classe Java anotada do JAX-WS gerado pelo Integration Designer para determinar o nome da operação de destino associada.

- Se a anotação @WebMethod estiver presente, o seletor de função usará a anotação @WebMethod para determinar o mapeamento de método Java correto para o método WSDL.
- Se a anotação @WebMethod estiver ausente, o seletor de função assumirá que o nome do método Java é o mesmo que o nome da operação chamada.

Nota: Este seletor de função é válido apenas para uma interface com tipo WSDL em uma importação EJB, não para uma interface com tipo Java em uma importação EJB.

O seletor de função retorna um objeto java.lang.reflect.Method que representa o método da interface EJB.

O seletor de função usa uma matriz de Objeto Java (Object[]) para conter a resposta do método de destino. O primeiro elemento no Object[] é um método Java com o nome do WSDL e o segundo elemento no Object[] é o objeto de negócios de entrada.

Seletor de Função de Exportação

Para processamento de entrada, o seletor de função de exportação deriva do método de destino a ser chamado a partir do método Java.

O seletor de função de exportação mapeia o nome da operação Java chamado pelo cliente EJB para o nome da operação na interface do componente de destino. O nome do método é retornado como uma Sequência e é resolvido pelo tempo de execução SCA dependendo do tipo de interface do componente de destino.

Ligações EIS

As ligações de Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Essa comunicação é obtida usando exportações EIS e importações EIS que suportam adaptadores de recursos JCA 1.5 e Websphere Adapters.

Seus componentes SCA podem requerer que dados sejam transferidos para ou a partir de um EIS externo. Ao criar um módulo SCA que requer tal conectividade, inclua (além de seu componente SCA) uma importação ou exportação com uma ligação EIS para comunicação com um EIS externo específico.

Adaptadores de recursos no IBM Integration Designer são usados dentro do contexto de uma importação ou uma exportação. Você desenvolve uma importação ou uma exportação com o assistente de serviço externo e, no seu desenvolvimento, inclui o adaptador de recursos. Uma importação EIS, que permite que seu aplicativo chame um serviço em um sistema EIS, ou uma exportação EIS, que permite que um aplicativo em um sistema EIS chame um serviço desenvolvido no IBM Integration Designer, é criada com um adaptador de recursos. Por exemplo, você poderia criar uma importação com o adaptador JD Edwards para chamar um serviço no sistema JD Edwards.

Quando você usa o assistente de serviço externo, as informações de ligação EIS são criadas para você. Também é possível usar uma outra ferramenta, o editor de montagem, para incluir ou modificar informações sobre ligação. Consulte *Acessando serviços externos com adaptadores* para obter mais informações.

Depois que o módulo SCA que contém a ligação EIS é implementado no servidor, é possível usar o console administrativo para visualizar informações sobre a ligação ou para configurar a ligação.

Visão Geral das Ligações EIS:

A ligação EIS (Enterprise Information System), quando usada com um adaptador de recursos JCA, permite acessar serviços em um Enterprise Information System ou tornar seus serviços disponíveis para o EIS.

O exemplo a seguir mostra como um módulo SCA denominado ContactSyncModule sincroniza informações de contato entre um sistema Siebel e um sistema SAP.

1. O componente SCA denominado ContactSync atende (por meio de uma exportação do aplicativo EIS denominada Contato Siebel) para alterações em contatos Siebel.
2. O componente SCA ContactSync em si faz uso de um aplicativo SAP (por meio de uma importação de aplicativo EIS) para atualizar as informações de contato SAP de acordo.

Como as estruturas de dados usadas para armazenar contatos são diferentes em sistemas Siebel e SAP, o componente SCA ContactSync deve fornecer mapeamento.

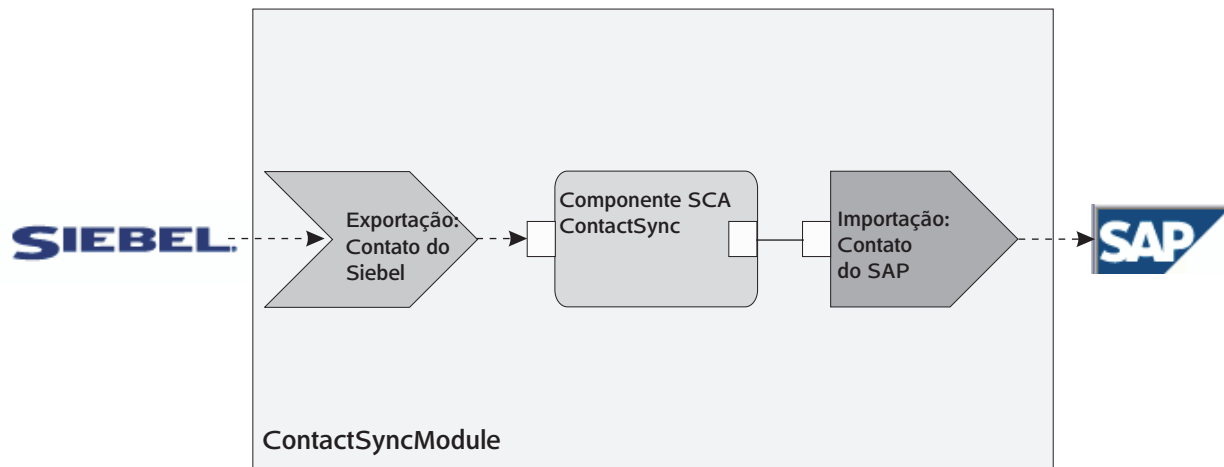


Figura 76. Fluxo a Partir de um Sistema Siebel para um Sistema SAP

A exportação de Contato Siebel e a importação de Contato SAP possuem os adaptadores de recursos apropriados configurados.

Recursos-chave de Ligações EIS:

Uma importação EIS é uma importação de Service Component Architecture (SCA) que permite que os componentes no módulo SCA usem aplicativos EIS definidos fora do módulo SCA. Uma importação EIS é usada para transferir os dados do componente SCA para um EIS externo; uma exportação EIS é usada para transferir dados de um EIS externo para o módulo SCA.

Importações

A função da importação EIS é ligar o intervalo entre os componentes SCA e sistemas EIS externos. Os aplicativos externos podem ser tratados como uma importação EIS. Nesse caso, a importação EIS envia dados para o EIS externo e, opcionalmente, recebe dados em resposta.

A importação EIS fornece componentes SCA com uma visualização uniforme dos aplicativos externos ao módulo. Isso permite que componentes se comuniquem com um EIS externo, tal como SAP, Siebel ou PeopleSoft, usando um modelo SCA consistente.

No lado do cliente da importação, há uma interface, exposta pelo aplicativo de importação EIS, com um ou mais métodos, cada um usando objetos de dados como argumentos e valores de retorno. No lado da implementação, há um Common Client Interface (CCI) implementado por um adaptador de recursos.

A implementação do tempo de execução de uma importação EIS conecta a interface do lado do cliente e o CCI. A importação mapeia a chamada do método na interface para a chamada no CCI.

As ligações são criadas em três níveis: a ligação de interface de ligação, que então usa as ligações de método contidas, que por sua vez usam ligações de dados.

A ligação de interface relaciona a interface da importação à conexão com o sistema EIS que fornece o aplicativo. Isto reflete o fato de que o conjunto de aplicativos, representado pela interface, é fornecido pela instância específica do EIS e a conexão oferece acesso a essa instância. O elemento de ligação contém propriedades com informações suficientes para criar a conexão (estas propriedades são parte da instância `javax.resource.spi.ManagedConnectionFactory`).

A ligação de método associa o método à interação específica com o sistema EIS. Para JCA, a interação é caracterizada pelo conjunto de propriedades da implementação da interface

`javax.resource.cci.InteractionSpec`. O elemento da interação da ligação de método contém estas propriedades, juntamente com o nome da classe, fornecendo, assim, informações suficientes para executar a interação. A ligação de método usa ligações de dados que descrevem o mapeamento do argumento e o resultado do método de interface para representação de EIS.

O cenário de tempo de execução para uma importação EIS é o seguinte:

1. O método na interface de importação é chamado usando o modelo de programação SCA.
2. A solicitação, que atinge a importação EIS, contém o nome do método e seus argumentos.
3. A importação primeiro cria uma implementação de ligação de interface; em seguida, usando dados da ligação de importação, cria uma `ConnectionFactory` e associa as duas. Ou seja, a importação chama `setConnectionFactory` na ligação da interface.
4. A implementação de ligação de método correspondente ao método chamado é criada.
5. A instância `javax.resource.cci.InteractionSpec` é criada e preenchida; em seguida, ligações de dados são usadas para ligar os argumentos do método com um formato entendido pelo adaptador de recursos.
6. A interface CCI é usada para executar a interação.
7. Quando a chamada retorna, a ligação de dados é usada para criar o resultado da chamada e o resultado é retornado para o responsável pela chamada.

Exportações

A função da exportação EIS é ligar o intervalo entre um componente SCA e um EIS externo. Aplicativos externos podem ser tratados como uma exportação EIS. Neste caso, o aplicativo externo envia seus dados na forma de notificações periódicas. Uma exportação EIS pode ser considerada como um aplicativo de assinatura atendendo a uma solicitação externa de um EIS. O componente SCA que usa a exportação EIS a visualiza como um aplicativo local.

A exportação EIS fornece aos componentes SCA uma visualização uniforme dos aplicativos externos ao módulo. Isto permite que componentes se comuniquem com um EIS, tal como SAP, Siebel ou PeopleSoft, usando um modelo SCA consistente.

A exportação apresenta uma implementação do listener recebendo solicitações do EIS. O listener implementa uma interface de listener específica do adaptador de recursos. A exportação também contém uma interface de implementação de componente, exposta ao EIS por meio da exportação.

A implementação do tempo de execução de uma exportação EIS conecta o listener à interface de implementação do componente. A exportação mapeia a solicitação de EIS para a chamada da operação apropriada no componente. As ligações são criadas em três níveis: a ligação de listener, que então usa a ligação de método nativa contida, que por sua vez usa uma ligação de dados.

A ligação do listener relaciona o listener que recebe solicitações com o componente exposto por meio da exportação. A definição de exportação contém o nome do componente; o tempo de execução o localiza e redireciona solicitações para ele.

A ligação de método nativo associa o método nativo ou o tipo de evento recebido pelo listener à operação implementada pelo componente exposto por meio da exportação. Não há relacionamento entre o método chamado no listener e o tipo de evento; todos os eventos chegam por meio de um ou mais métodos do listener. A ligação de método nativo usa o seletor de função definido na exportação para extrair o nome do método nativo a partir dos dados de entrada e ligações de dados para ligar o formato de dados do EIS em um formato entendido pelo componente.

O cenário de tempo de execução para uma exportação EIS é o seguinte:

1. A solicitação de EIS aciona a chamada do método na implementação do listener.
2. O listener localiza e chama a exportação, transmitindo para ela todos os argumentos de chamada.

3. A exportação cria a implementação de ligação do listener.
4. A exportação instancia o seletor de função e o configura na ligação do listener.
5. A exportação inicializa as ligações de método nativo e as inclui na ligação do listener. Para cada ligação de método nativo, as ligações de dados também são inicializadas.
6. A exportação chama a ligação do listener.
7. A ligação do listener localiza componentes exportados e usa o seletor de função para recuperar o nome do método nativo.
8. Este nome é usado para localizar a ligação de método nativo que, então, chama o componente de destino.

O estilo de interação do adaptador permite que a ligação de exportação do EIS chame o componente de destino assincronicamente (o padrão) ou de forma síncrona.

Adaptadores de Recursos

Você desenvolve uma importação ou uma exportação com o assistente de serviço externo e, ao desenvolvê-la, inclui um adaptador de recursos. Os adaptadores que vêm com IBM Integration Designer usados para acessar sistemas Siebel, SAP, PeopleSoft, JD Edwards, IMS e CICS são destinados apenas para propósitos de desenvolvimento e teste. Isso significa que são usados para desenvolver e testar seus aplicativos.

Quando implementar seu aplicativo, adaptadores de tempo de execução licenciados para executar seu aplicativo serão necessários. No entanto, quando constrói seu serviço, é possível incorporar o adaptador com seu serviço. O licenciamento de seu adaptador pode permitir que o adaptador integrado seja usado como o adaptador de tempo de execução licenciado. Estes adaptadores são compatíveis com o Java EE Connector Architecture (JCA 1.5). O JCA, um padrão aberto, é o padrão de Java EE para conectividade do EIS. O JCA fornece uma estrutura gerenciada; isto é, a Qualidade do Serviço (QoS) é fornecida pelo servidor de aplicativos, que oferece gerenciamento de ciclo de vida e segurança a transações. Eles também são compatíveis com a especificação do Enterprise Metadata Discovery, com a exceção do IBM CICS ECI Resource Adapter e do IBM IMS Connector para Java.

Os WebSphere Business Integration Adapters, um conjunto mais antigo de adaptadores, também são suportados pelo assistente.

Recursos Java EE

O módulo EIS, um módulo SCA que segue o padrão de módulo EIS, pode ser implementado na plataforma Java EE.

A implementação do módulo EIS na plataforma Java EE resulta em um aplicativo que está pronto para iniciar, compactado como um arquivo EAR e implementado no servidor. Todos os artefatos Java EE e recursos são criados; o aplicativo está configurado e pronto para ser executado.

Propriedades Dinâmicas de Especificação de Interação JCA e de Especificação de Conexão:

A ligação EIS pode aceitar a entrada para InteractionSpec e ConnectionSpec especificadas usando um objeto de dados filho bem definido que acompanha a carga útil. Isso permite interações de solicitação-resposta dinâmicas com um adaptador de recursos por meio de InteractionSpec e autenticação do componente por meio de ConnectionSpec.

`javax.cci.interactionspec` transporta informações sobre como a solicitação de interação com o adaptador de recursos deve ser manipulada. Ela também pode transportar informações sobre como a interação foi obtida após a solicitação. Essas comunicações bidirecionais através das interações às vezes são referidas como *conversas*.

A ligação EIS espera que a carga útil que será o argumento para o adaptador de recursos contenha um objeto de dados filho chamado **properties**. Este objeto de dados de propriedade conterá pares nome/valor, com o nome das propriedades de Especificação da Interação em um formato específico. As regras de formatação são:

- Os nomes devem iniciar com o prefixo **IS**, seguido pelo nome da propriedade. Por exemplo, uma especificação de interação com uma propriedade JavaBeans chamada **InteractionId** especificaria o nome da propriedade como **ISInteractionId**.
- O par nome/valor representa o nome e o valor do tipo simples da propriedade de Especificação de Interação.

Neste exemplo, uma interface específica que a entrada de uma operação é um objeto de dados **Account**. Essa interface chama um aplicativo de ligação de importação EIS com a intenção de enviar e receber uma propriedade dinâmica **InteractionSpec** chamada **workingSet** com o valor **xyz**.

O gráfico de negócios ou os objetos de negócios no servidor contêm um objeto de negócios **properties** subjacente que permite o envio de dados específicos de protocolo com a carga útil. Este objeto de negócios **properties** é integrado e não precisa ser especificado no esquema XML ao construir um objeto de negócios. Ele precisa apenas ser criado e usado. Se você tiver seus próprios tipos de dados definidos com base em um esquema XML, será necessário especificar um elemento **properties** que contém seus pares nome/valor esperados.

```
BOFactory dataFactory = (BOFactory) \
serviceManager.locateService("com/ibm/websphere/bo/BOFactory");
//Wrapper for doc-lit wrapped style interfaces,
//skip to payload for non doc-lit
DataObject docLitWrapper = dataFactory.createElement /
("http://mytest/eis/Account", "AccountWrapper");
```

Crie a carga útil.

```
DataObject account = docLitWrapper.createDataObject(0);
DataObject accountInfo = account.createDataObject("AccountInfo");
//Perform your setting up of payload
```

```
//Construct properties data for dynamic interaction
```

```
DataObject properties = account.createDataObject("properties");
```

Para o nome **workingSet**, configure o valor esperado (**xyz**).

```
properties.setString("ISworkingSet", "xyz");
```

```
//Invoke the service with argument
```

```
Service accountImport = (Service) \
serviceManager.locateService("AccountOutbound");
DataObject result = accountImport.invoke("createAccount", docLitWrapper);
```

```
//Get returned property
```

```
DataObject retProperties = result.getDataObject("properties");
```

```
String workingset = retProperties.getString("ISworkingSet");
```

É possível usar propriedades **ConnectionSpec** para autenticação do componente dinâmico. As mesmas regras se aplicam conforme acima, exceto que o prefixo do nome da propriedade precisa ser **CS** (em vez de **IS**). As propriedades **ConnectionSpec** não são bidirecionais. O mesmo objeto de dados **properties** pode conter propriedades **IS** e **CS**.

Para usar propriedades ConnectionSpec, configure o **resAuth** especificado na ligação de importação como **Application**. Além disso, certifique-se de que o adaptador de recursos suporte autorização de componente. Consulte o capítulo 8 da Especificação J2EE Connector Architecture para obter mais detalhes.

Clientes Externos com Ligações EIS:

O servidor pode enviar mensagens para, ou receber mensagens de, clientes externos usando ligações EIS.

Um cliente externo, por exemplo, um portal da web ou um EIS, precisa enviar uma mensagem a um módulo SCA no servidor ou precisa ser chamado por um componente a partir do servidor.

O cliente chama a importação EIS como com qualquer outro aplicativo, usando a Interface de Chamada Dinâmica (DII) ou interface Java.

1. O cliente externo cria uma instância do ServiceManager e consulta a importação EIS usando seu nome de referência. O resultado da consulta é uma implementação da interface de serviço.
2. O cliente cria um argumento de entrada, um objeto de dados genérico, criado dinamicamente usando o esquema de objeto de dados. Esta etapa é feita usando a implementação da interface DataFactory do Objetos de Dados de Serviço.
3. O cliente externo chama o EIS e obtém os resultados requeridos.

Como alternativa, o cliente pode chamar a importação EIS usando a interface Java.

1. O cliente cria uma instância do ServiceManager e consulta a importação de EIS usando seu nome de referência. O resultado da consulta é uma interface Java da importação EIS.
2. O cliente cria um argumento de entrada e um objeto de dados digitado.
3. O cliente chama o EIS e obtém os resultados necessários.

A interface de exportação EIS define a interface do componente SCA exportado que está disponível para os aplicativos EIS externos. Essa interface pode ser considerada como a interface que um aplicativo externo (como SAP ou PeopleSoft) chamará por meio da implementação do tempo de execução do aplicativo de exportação EIS.

A exportação usa EISExportBinding para ligar serviços exportados para o aplicativo EIS externo. Ela permite subscrever um aplicativo contido em seu módulo SCA para atender solicitações de serviço EIS. A ligação de exportação EIS especifica o mapeamento entre a definição de eventos de entrada conforme ele é entendido pelo adaptador de recursos (usando interfaces Java EE Connector Architecture) e a chamada de operações SCA.

EISExportBinding requer que serviços EIS externos sejam baseado em contratos de entrada Java EE Connector Architecture 1.5. EISExportBinding requer que um manipulador de dados ou uma ligação de dados seja especificada no nível de ligação ou no nível de método.

Ligações JMS

Um provedor de Sistema de Mensagens Java (JMS) ativa o sistema de mensagens com base na API e no modelo de programação do Serviço de Sistema de Mensagens Java. Ele fornece connection factories JMS para criar conexões com destinos JMS e enviar e receber mensagens.

Ligações JMS podem ser usadas ao interagir com a ligação de provedor Barramento de Integração de Serviços (SIB) e são compatíveis com JMS e JCA 1.5.

É possível usar as ligações de exportação e importação JMS um módulo de Service Component Architecture (SCA) para fazer chamadas e receber mensagens de sistemas externos JMS.

As ligações de importação e exportação JMS fornecem integração com aplicativos JMS usando o provedor JMS SIB baseado em JCA 1.5 que faz parte do WebSphere Application Server. Outros adaptadores de recursos JMS baseados em JCA 1.5 não são suportados

Visão Geral de Ligações JMS:

As ligações JMS fornecem conectividade entre o ambiente de Service Component Architecture (SCA) e sistemas JMS.

Ligações JMS

Os principais componentes das ligações de importação JMS e de exportação JMS são:

- adaptador de Recursos: permite a conectividade bidirecional gerenciada entre um módulo SCA e sistemas JMS externos
- Conexões: Encapsular uma conexão virtual entre um cliente e um aplicativo de provedor
- Destinos: usado por um cliente para especificar o destino de mensagens que ele produz ou a origem das mensagens que ele consome
- Dados de Autenticação: usado para proteger o acesso à ligação

Recursos-chave de Ligações JMS

Cabeçalhos Especiais

Propriedades de cabeçalho especiais são usadas em importações e exportações JMS para informar ao destino como manipular a mensagem.

Por exemplo, TargetFunctionName mapeia do método nativo para o método de operação.

Recursos Java EE

Um número de recursos Java EE é criado quando importações e exportações do JMS são implementados em um ambiente Java EE.

ConnectionFactory

Usado pelos clientes para criar uma conexão com o provedor JMS.

ActivationSpec

As importações usam isso para receber a resposta para uma solicitação; exportações o usam ao configurar os terminais de mensagem que representam listeners de mensagens em suas interações com o sistema de mensagens.

Destinos

- Destino de envio: em uma importação, este é o local onde a solicitação ou mensagem de saída é enviada; em uma exportação, este é o destino onde a mensagem de resposta será enviada, se não substituído pelo campo de cabeçalho JMSReplyTo na mensagem recebida.
- Destino de recebimento: onde a mensagem recebida deve ser colocada; com importações, esta é uma resposta; com exportações, esta é uma solicitação.
- Destino de retorno de chamada: Destino do sistema JMS de SCA usado para armazenar informações de correlação. Não leia ou grave neste destino.

A tarefa de instalação cria ConnectionFactory e três destinos. Também cria ActivationSpec para ativar o listener de mensagem de tempo de execução para atender respostas no destino de recebimento. As propriedades desses recursos são especificadas no arquivo de importação ou exportação.

Integração de JMS e Adaptadores de Recursos:

O Sistema de Mensagens Java (JMS) fornece integração através de um adaptador de recursos baseado em JMS JCA 1.5 disponível. O suporte completo para integração de JMS é fornecido para o adaptador de recursos do Barramento de Integração de Serviços (SIB) JMS.

Use um provedor JMS para o adaptador de recursos JCA 1.5 quando desejar integrar com um sistema JMS compatível com JCA 1.5 externo. Serviços externos compatíveis com JCA 1.5 podem receber mensagens e enviar mensagens para integrar com seus componentes Service Component Architecture (SCA) usando o adaptador de recursos SIB JMS.

O uso de outros adaptadores de recursos JCA 1.5 específicos do provedor não é suportado.

Ligações de Importação e Exportação JMS:

É possível fazer módulos SCA interagirem com serviços fornecidos por aplicativos JMS externos usando ligações de importação e exportação JMS.

Ligações de Importação JMS

Conexões com o provedor JMS associado de destinos JMS são criadas usando uma connection factory JMS. Use objetos administrativos de connection factory para gerenciar connection factories JMS para o provedor de sistemas de mensagens padrão.

Interação com sistemas JMS externos inclui o uso de destinos para enviar solicitações e receber respostas.

Dois tipos de cenários de uso para a ligação de importação JMS são suportados, dependendo do tipo de operação que está sendo chamado:

- Unidirecional: A importação JMS coloca uma message no destino de envio configurado na ligação de importação. Nada é configurado no campo replyTo do cabeçalho JMS.
- Bidirecional (Solicitação-Resposta): A importação JMS coloca uma mensagem no destino de envio e, em seguida, persiste a resposta que recebe do componente SCA.

A ligação de importação pode ser configurada (usando o campo **Esquema de Correlação de Resposta** no Integration Designer) para esperar que o ID de correlação de mensagem de resposta tenha sido copiado do ID de mensagem de solicitação (o padrão) ou do ID de correlação da mensagem de solicitação. A ligação de importação também pode ser configurada para usar um destino de resposta dinâmica temporária para correlacionar respostas com solicitações. Um destino temporário é criado para cada solicitação e a importação usa este destino para receber a resposta.

O destino de recebimento é configurado na propriedade do cabeçalho replyTo da mensagem de saída. Um listener de mensagem é implementado para atender o destino de recebimento e, quando uma resposta é recebida, o listener de mensagem transmite a resposta de volta para o componente.

Para ambos os cenários de uso, unidirecional e bidirecional, propriedades de cabeçalho dinâmicas e estáticas podem ser especificadas. As propriedades estáticas podem ser configuradas a partir da ligação de método de importação JMS. Algumas destas propriedades tem significados especiais para o tempo de execução JMS SCA.

É importante observar que JMS é uma ligação assíncrona. Se um componente de chamada chamar uma importação JMS sincronicamente (para uma operação bidirecional), o componente de chamada será bloqueado até que a resposta seja retornada pelo serviço JMS.

Figura 32 na página 121 ilustra como a importação está vinculada ao serviço externo.

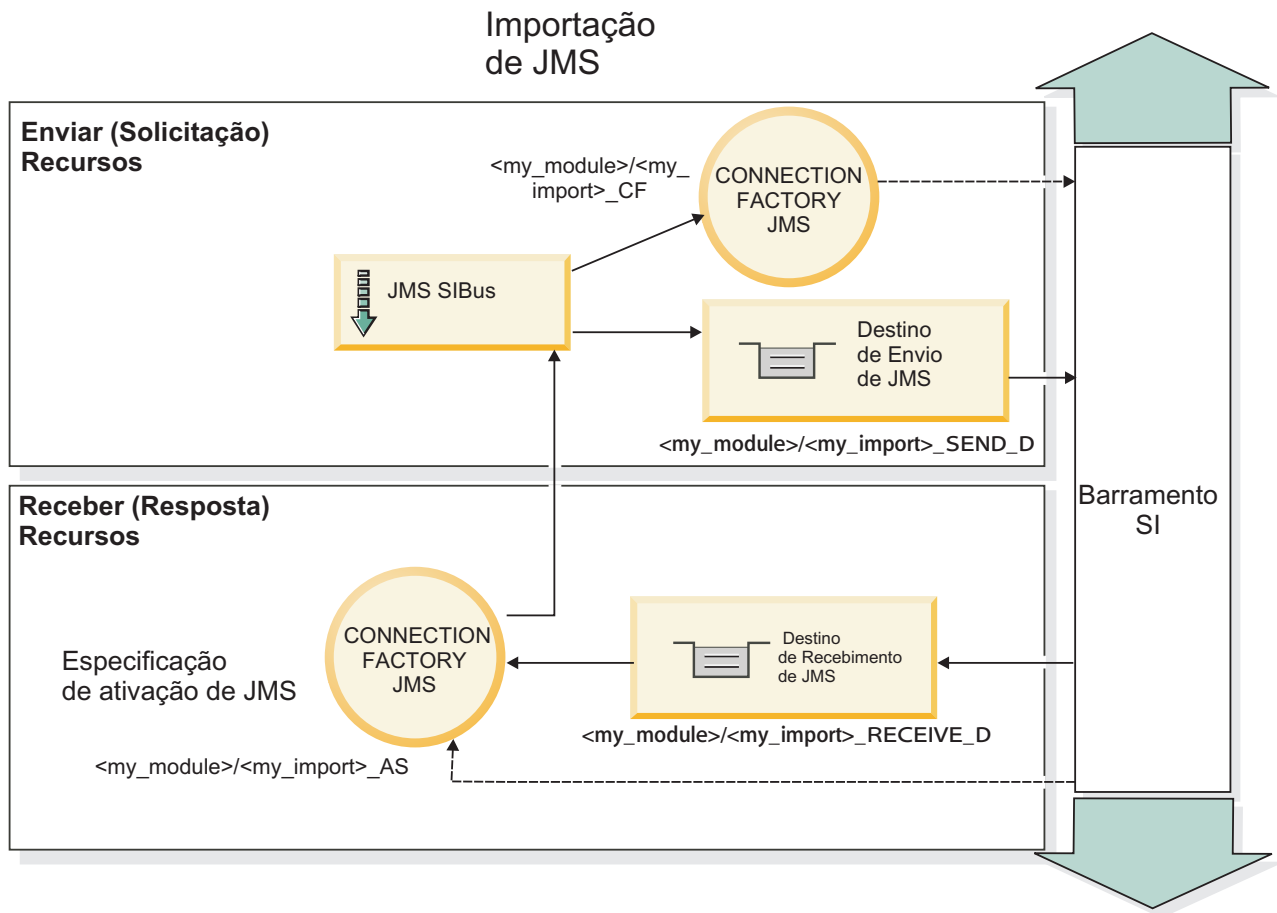


Figura 77. Recursos de Ligação de Importação JMS

Ligações de Exportação JMS

As ligações de exportação JMS fornecem os meios para módulos SCA fornecerem serviços para aplicativos JMS externos.

A conexão que faz parte de uma exportação JMS é uma especificação de ativação configurável.

Uma exportação JMS possui destinos de envio e recebimento.

- O destino de recebimento é onde a mensagem recebida para o componente de destino deve ser colocada.
- O destino de envio é onde a resposta será enviada, a menos que a mensagem recebida tenha substituído isto usando a propriedade de cabeçalho `replyTo`.

Um listener de mensagem é implementado para atender solicitações recebidas no destino de recebimento especificado na ligação de exportação. O destino especificado no campo Enviar é utilizado para enviar a resposta para a solicitação de entrada se o componente chamado fornecer uma resposta. O destino especificado no campo `replyTo` da mensagem recebida substitui o destino especificado no envio.

O Figura 33 na página 122 ilustra como o solicitante externo está vinculado à exportação.

Exportação de JMS

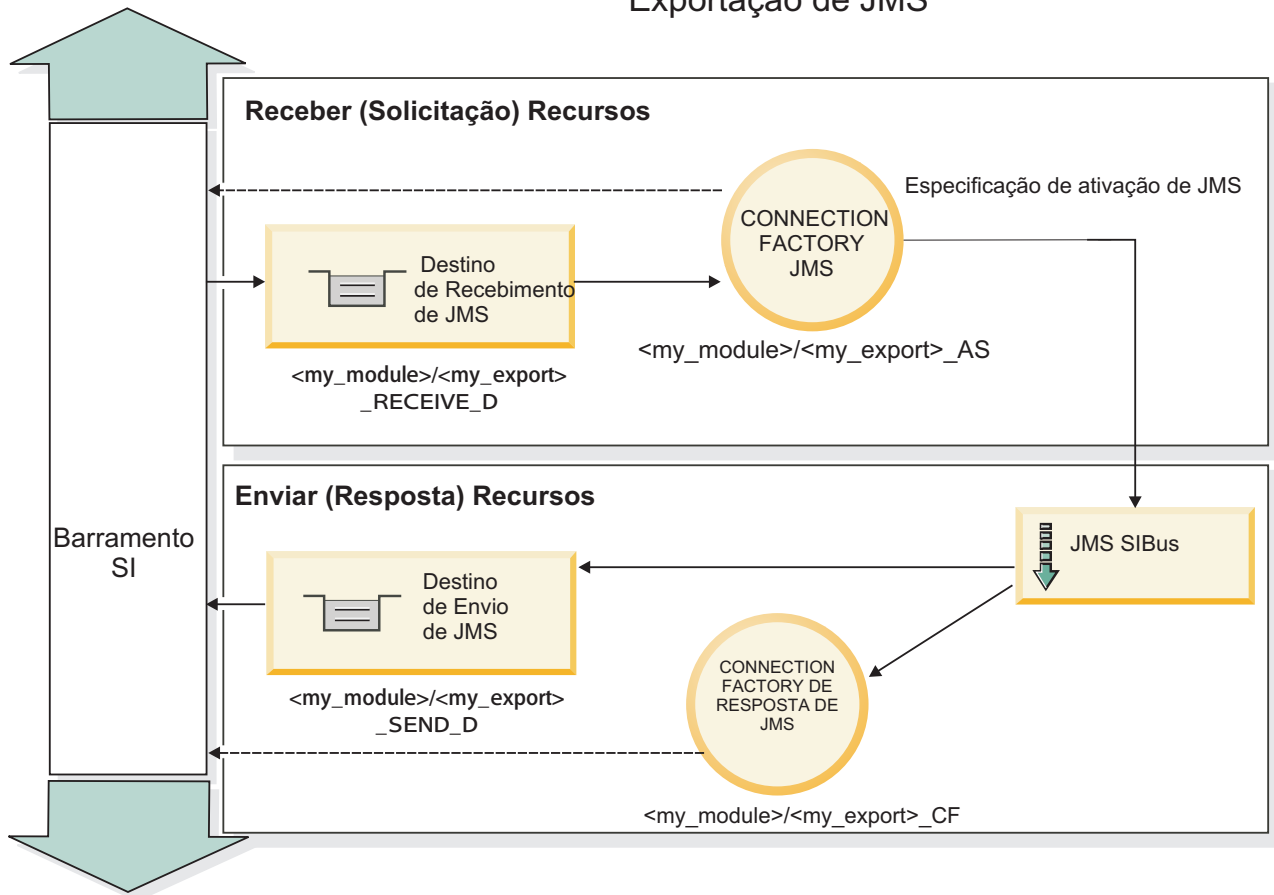


Figura 78. Recursos de Ligação de Exportação JMS

Cabeçalhos JMS:

Uma mensagem JMS contém dois tipos de cabeçalhos - o cabeçalho do sistema JMS e diversas propriedades JMS. Ambos os tipos de cabeçalhos podem ser acessados em um módulo de mediação no Objeto de Mensagem de Serviço (SMO) ou usando a API ContextService.

Cabeçalho do Sistema JMS

O cabeçalho do sistema JMS é representado no SMO pelo elemento JMSHeader, que contém todos os campos geralmente localizados em um cabeçalho JMS. Embora eles possam ser modificados na mediação (ou ContextService), alguns campos de cabeçalho do sistema JMS configurados no SMO não serão propagados na mensagem JMS de saída pois eles são substituídos pelo sistema ou por valores estáticos.

Os campos-chave no cabeçalho do sistema JMS que podem ser atualizados em uma mediação (ou ContextService) são:

- **JMSType** e **JMSCorrelationID** - valores das propriedades de cabeçalho de mensagem predefinidas específicas
- **JMSDeliveryMode** - valores para modo de entrega (persistente ou não persistente; o padrão é persistente)
- **JMSPriority** - valor de prioridade (0 a 9; padrão é JMS_Default_Priority)

Propriedades JMS

As propriedades JMS são representadas no SMO como entradas na lista de Propriedades. As propriedades podem ser incluídas, atualizadas ou excluídas em uma mediação ou usando a API ContextService.

As propriedades também podem ser configuradas estaticamente na ligação JMS. Propriedades que são configuradas estaticamente substituem as configurações (com o mesmo nome) que são configuradas dinamicamente.

As propriedades do usuário propagadas a partir de outras ligações (por exemplo, uma ligação HTTP) serão enviadas na ligação JMS como propriedades JMS.

Configurações de Propagação de Cabeçalho

A propagação do cabeçalho do sistema JMS e das propriedades a partir da mensagem JMS de entrada para componentes de recebimento de dados ou de componentes de envio de dados para a mensagem JMS de saída pode ser controlada pelo sinalizador Propagar Cabeçalho do Protocolo na ligação.

Quando Propagar Cabeçalho de Protocolo é configurado, as informações de cabeçalho têm permitidas para fluírem para a mensagem ou para o componente de destino, conforme descrito na lista a seguir:

- Solicitação de exportação JMS
O cabeçalho JMS recebido na mensagem será propagado para os componentes de destino por meio do serviço de contexto. As propriedades JMS recebidas na mensagem serão propagadas para componentes de destino por meio do serviço de contexto.
- Resposta de exportação JMS
Quaisquer campos de cabeçalho JMS configurados no serviço de contexto serão usados na mensagem de saída, se não substituídos por propriedades estáticas configuradas na ligação de exportação JMS. Quaisquer propriedades configuradas no serviço de contexto serão usadas na mensagem de saída se não substituídas por propriedades estáticas configuradas na ligação de exportação JMS.
- Solicitação de importação JMS
Quaisquer campos de cabeçalho JMS configurados no serviço de contexto serão usados na mensagem de saída, se não substituídos por propriedades estáticas configuradas na ligação de importação JMS. Quaisquer propriedades configuradas no serviço de contexto serão usadas na mensagem de saída se não substituídas por propriedades estáticas configuradas na ligação de importação JMS.
- Resposta de importação JMS
O cabeçalho JMS recebido na mensagem será propagado para os componentes de destino por meio do serviço de contexto. As propriedades JMS recebidas na mensagem serão propagadas para componentes de destino por meio do serviço de contexto.

Esquema de correlação de destino de resposta dinâmica temporária do JMS:

O esquema de correlação de destino de resposta dinâmica temporário faz com que uma fila dinâmica ou tópico exclusivo seja criado para cada solicitação enviada.

O destino da resposta estática especificado na importação é usado para derivar a natureza da fila de destino dinâmica temporária ou tópico. Isto é configurado no campo **ReplyTo** da solicitação e a importação JMS atende respostas nesse destino. Quando a resposta é recebida, ela é reenfileirada para o destino de resposta estática para processamento assíncrono. O campo **CorrelationID** da resposta não é usado e não precisa ser configurado.

Problemas Transacionais

Quando um destino dinâmico temporário está sendo usado, a resposta deve ser consumida no mesmo encadeamento que a resposta enviada. A solicitação deve ser enviada fora da transação global e deve ser confirmada antes de ser recebida pelo serviço de backend e uma resposta retornada.

Persistência

As filas dinâmicas temporárias são entidades de curta duração e não garantem o mesmo nível de persistência associado a uma fila ou tópico estático. Uma fila dinâmica temporária ou tópico não sobreviverá a uma reinicialização de servidor e nem as mensagens. Após a mensagem ter sido reenqueueada no destino de resposta estática, ela retém a persistência definida na mensagem.

Tempo Limite

A importação aguarda para receber a resposta no destino de resposta dinâmica temporário para um período de tempo fixo. Este intervalo de tempo será obtido do qualificador de Prazo de Expiração de Resposta de SCA, se ele estiver configurado, caso contrário, o tempo padrão será 60 segundos. Se o tempo de espera for excedido, a importação lançará uma `IServiceTimeoutRuntimeException`.

Clientes Externos:

O servidor pode enviar mensagens para, ou receber mensagens de, clientes externos usando ligações JMS.

Um cliente externo (como um portal da web ou um Enterprise Information System) pode enviar uma mensagem para um módulo SCA no servidor ou pode ser chamado por um componente a partir do servidor.

Os componentes de exportação JMS implementam listeners de mensagem para atender solicitações recebidas no destino de recebimento especificado na ligação de exportação. O destino especificado no campo de envio é usado para enviar a resposta para a solicitação de entrada se o aplicativo chamado fornece uma resposta. Portanto, um cliente externo é capaz de chamar aplicativos com a ligação de exportação.

As importações JMS interagem com clientes externos enviando mensagens e recebendo mensagens das filas JMS.

Trabalhando com Clientes Externos:

Um cliente externo (ou seja, fora do servidor) pode precisar interagir com um aplicativo instalado no servidor.

Considere um cenário muito simples no qual um cliente externo deseja interagir com um aplicativo no servidor. A figura descreve um cenário simples típico.

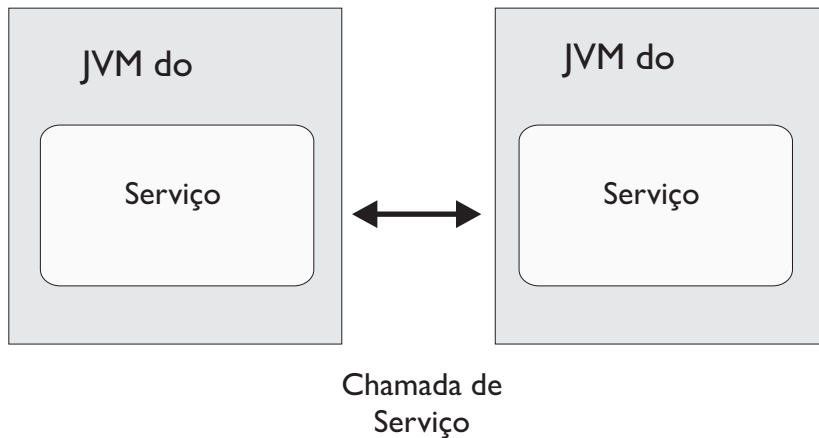


Figura 79. Cenário de Caso de Uso Simples: Cliente Externo Interage com o Aplicativo do Servidor

O aplicativo SCA inclui uma exportação com uma ligação JMS; isso torna o aplicativo disponível para clientes externos.

Quando você tem um cliente externo em uma Java Virtual Machine (JVM) separada de seu servidor, há várias etapas que você deve executar para fazer uma conexão e interagir com uma exportação de JMS. O cliente obtém um InitialContext com os valores corretos e, em seguida, consulta os recursos através de JNDI. O cliente, então, usa o cliente de especificação JMS 1.1 para acessar os destinos e as mensagens de envio e recebimento nos destinos.

Os nomes JNDI padrão dos recursos criados automaticamente pelo tempo de execução são listados no tópico de configuração desta seção. No entanto, se você tiver recursos pré-criados, use esses nomes JNDI.

1. Configure os destinos JMS e a connection factory para enviar a mensagem.
2. Certifique-se de que o contexto JNDI, a porta para o adaptador de recursos do SIB e a porta de autoinicialização do sistema de mensagens estejam corretos.

O servidor usa algumas portas padrão mas, se houver mais servidores instalados nesse sistema, portas alternativas serão criadas no momento da instalação para evitar conflitos com outras instâncias do servidor. É possível usar o console administrativo para determinar quais portas seu servidor está utilizando. Acesse **Servidores > Servidores de Aplicativos > your_server_name > Configuração** e clique em **Portas** em **Comunicação**. É possível, então, editar a porta que está sendo usada.

3. O cliente obtém um contexto inicial com os valores corretos e, em seguida, consulta os recursos através de JNDI.
4. Usando especificações JMS 1.1, o cliente acessa os destinos e as mensagens de envio e recebimento nos destinos.

Resolução de Problemas de Ligações JMS:

É possível diagnosticar e corrigir problemas com ligações JMS.

Exceções de Implementação

Em resposta a várias condições de erro, a implementação de importação e exportação de JMS pode retornar um de dois tipos de exceções:

- Exceção de Negócios de Serviço: esta exceção é retornada se a falha especificada na interface de negócios de serviço (tipo de porta WSDL) ocorreu.
- Exceção de Tempo de Execução de Serviço: levantada em todos os outros casos. Na maioria dos casos, a exceção de causa conterá a exceção original (JMSException).

Por exemplo, uma importação espera apenas uma mensagem de resposta para cada mensagem de solicitação. Se mais de uma resposta chegar, ou se uma resposta atrasada (uma para a qual a expiração de resposta de SCA expirou) chegar, uma Exceção de Tempo de Execução de Serviço será emitida. A transação é retrocedida e a mensagem de resposta é retirada da fila ou manipulada pelo gerenciador de eventos com falha.

Principais Condições de Falha

As principais condições de falha das ligações JMS são determinadas pela semântica transacionais, pela configuração do provedor JMS ou pela referência ao comportamento existente em outros componentes. As principais condições de falha incluem:

- Falha ao se conectar ao provedor JMS ou destino.

Uma falha de conexão com o provedor JMS para receber mensagens resultará na falha do listener de mensagem ao iniciar. Essa condição será registrada no log do WebSphere Application Server. As mensagens persistentes permanecerão no destino até que elas sejam recuperadas com sucesso (ou expiradas).

Uma falha de conexão com o provedor JMS para enviar mensagens de saída resultará no retrocesso da transação que controla o envio.

- Falha ao analisar uma mensagem de entrada ou ao construir uma mensagem de saída.

Uma falha na ligação de dados ou no manipulador de dados causa o retrocesso da transação que controla o trabalho.

- Falha ao enviar a mensagem de saída.

Uma falha ao enviar uma mensagem causa o retrocesso da transação relevante.

- Mensagens de resposta atrasadas diversas ou inesperadas.

A importação espera apenas uma mensagem de resposta para cada mensagem de solicitação. Além disso, o período de tempo válido no qual uma resposta pode ser recebida é determinado pelo qualificador de Expiração de Resposta SCA na solicitação. Quando uma resposta chega ou o tempo de expiração é excedido, o registro de correlação é excluído. Se mensagens de resposta chegarem inesperadamente ou chegarem atrasadas, uma Exceção de Tempo de Execução de Serviço será emitida.

- Exceção de tempo de execução de tempo limite de serviço causada por resposta atrasada ao usar o esquema de correlação de destino de resposta dinâmica temporário.

A importação de JMS expirará após um período de tempo determinado pelo qualificador de expiração de resposta de SCA ou se, se isto não for configurado, será padronizado como 60 segundos.

Mensagens SCA Baseadas em JMS que Não Aparecem no Gerenciador de Eventos com Falha

Se mensagens SCA originadas por uma interação do JMS falharem, você pode encontrar essas mensagens no gerenciador de eventos com falha. Se tais mensagens não estiverem aparecendo no gerenciador de eventos com falha, assegure-se de que o destino de SIB subjacente do destino JMS tenha um valor máximo de entregas com falha maior que **1**. Configurar esse valor como **2** ou mais permite a interação com o gerenciador de eventos com falha durante chamadas de SCA para as ligações JMS.

Exceções de Manipulação:

A maneira na qual a ligação é configurada determina como as exceções que são levantadas por manipuladores de dados ou ligações de dados são manipuladas. Além disso, a natureza do fluxo de mediação determina o comportamento do sistema quando uma exceção desse tipo é lançada.

Uma variedade de problemas pode ocorrer quando um manipulador de dados ou uma ligação de dados é chamada por sua ligação. Por exemplo, um manipulador de dados pode receber uma mensagem que possui uma carga útil corrompida ou pode tentar ler uma mensagem que possua um formato incorreto.

A maneira que sua ligação trata uma exceção desse tipo é determinada por como você implementa o manipulador de dados ou a ligação de dados. O comportamento recomendado é que você projete sua ligação de dados para lançar uma **DataBindingException**.

Quando qualquer exceção de tempo de execução, incluindo uma **DataBindingException**, é lançada:

- Se o fluxo de mediação for configurado para ser transacional, a Mensagem JMS, por padrão, será armazenada no Gerenciador de Evento com Falha para exclusão ou reprodução manual.

Nota: É possível alterar o modo de recuperação na ligação de modo que a mensagem seja retrocedida ao invés de ser armazenada no Gerenciador de Evento com Falha.

- Se o fluxo de mediação não for transacional, a exceção será registrada e a mensagem será perdida.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, qualquer exceção do manipulador de dados é agrupada em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Ligações JMS Genéricas

A ligação JMS genérica fornece conectividade com provedores compatíveis com JMS 1.1 de terceiros. A operação das ligações JMS Genéricas é semelhante àquela de ligações JMS.

O serviço fornecido por meio de uma ligação JMS permite que um módulo Service Component Architecture (SCA) faça chamadas ou receba mensagens de sistemas externos. O sistema pode ser um sistema JMS externo.

A Ligação JMS Genérica fornece integração com provedores JMS não compatíveis com JCA 1.5 que suportam JMS 1.1 e implementam o Application Server Facility do JMS opcional. A ligação JMS Genérica suporta esses provedores JMS (incluindo Oracle AQ, TIBCO, SonicMQ, WebMethods e BEA WebLogic) que não suportam JCA 1.5, mas suporta o Application Server Facility da especificação JMS 1.1. O provedor JMS integrado pelo WebSphere (SIBJMS), que é um provedor JMS JCA 1.5, não é suportado por esta ligação; ao usar esse provedor, use o “Ligações JMS” na página 118.

Use esta ligação Genérica durante a integração com um sistema baseado em JMS não compatível com JCA 1.5 em um ambiente SCA. Os aplicativos externos de destino podem, então, receber mensagens e enviar mensagens para integrar com um componente SCA.

Visão Geral de Ligações JMS Genéricas:

Ligações JMS genéricas são ligações JMS não JCA que fornecem conectividade entre o ambiente Service Component Architecture (SCA) e sistemas de JMS que estão em conformidade com o JMS 1.1 e que implementam o Application Server Facility do JMS opcional.

Ligações JMS Genéricas

Os principais aspectos de ligações de importação e exportação do JMS Genérico incluem o seguinte:

- Porta Listener: permite que os provedores JMS não baseados em JCA recebam mensagens e as despachem para um Bean Acionado por Mensagens (MDB)
- Conexões: Encapsular uma conexão virtual entre um cliente e um aplicativo de provedor
- Destinos: usado por um cliente para especificar o destino de mensagens que ele produz ou a origem das mensagens que ele consome
- Dados de Autenticação: usado para proteger o acesso à ligação

Ligações de Importação JMS Genéricas

As ligações de importação do JMS genérico permitem que componentes dentro do módulo SCA se comuniquem com serviços fornecidos pelos provedores externos JMS não compatíveis com JCA 1.5.

A parte de conexão de uma importação de JMS é um connection factory. Uma connection factory, o objeto que um cliente usa para criar uma conexão com um provedor, encapsula um conjunto de parâmetros de configuração de conexão definidos por um administrador. Cada connection factory é uma instância da interface ConnectionFactory, QueueConnectionFactory ou TopicConnectionFactory.

Interação com sistemas JMS externos inclui o uso de destinos para enviar solicitações e receber respostas.

Dois tipos de cenários de uso para a ligação de importação do JMS Genérico são suportados, dependendo do tipo de operação que está sendo chamado:

- Unidirecional: A importação do JMS Genérico coloca uma mensagem no destino de envio configurado na ligação de importação. Nada é enviado para o campo replyTo do cabeçalho JMS.
- Bidirecional (Solicitação-Resposta): A importação do JMS Genérico coloca uma mensagem no destino de envio e, em seguida, persiste a resposta que recebe do componente SCA.

O destino de recebimento é configurado na propriedade do cabeçalho replyTo da mensagem de saída. Um bean acionado por mensagens (MDB) é implementado para atender no destino de recebimento e, quando uma resposta é recebida, o MDB transmite a resposta de volta para o componente.

A ligação de importação pode ser configurada (utilizando o campo **Esquema de Correlação de Resposta** no Integration Designer) para esperar que o ID de correlação da mensagem de resposta tenha sido copiado do ID de mensagem de solicitação (o padrão) ou a partir do ID de correlação da mensagem de solicitação.

Para ambos os cenários de uso, unidirecional e bidirecional, propriedades de cabeçalho dinâmicas e estáticas podem ser especificadas. As propriedades estáticas podem ser configuradas a partir da ligação do método de importação do JMS Genérico. Algumas destas propriedades tem significados especiais para o tempo de execução JMS SCA.

É importante observar que JMS Genérica é uma ligação assíncrona. Se um componente de chamada chamar uma importação JMS Genérica sincronicamente (para uma operação bidirecional), o componente de chamada será bloqueado até a resposta ser retornada pelo serviço JMS.

Figura 35 na página 129 ilustra como a importação está vinculada ao serviço externo.

(Genérico) Importação de JMS

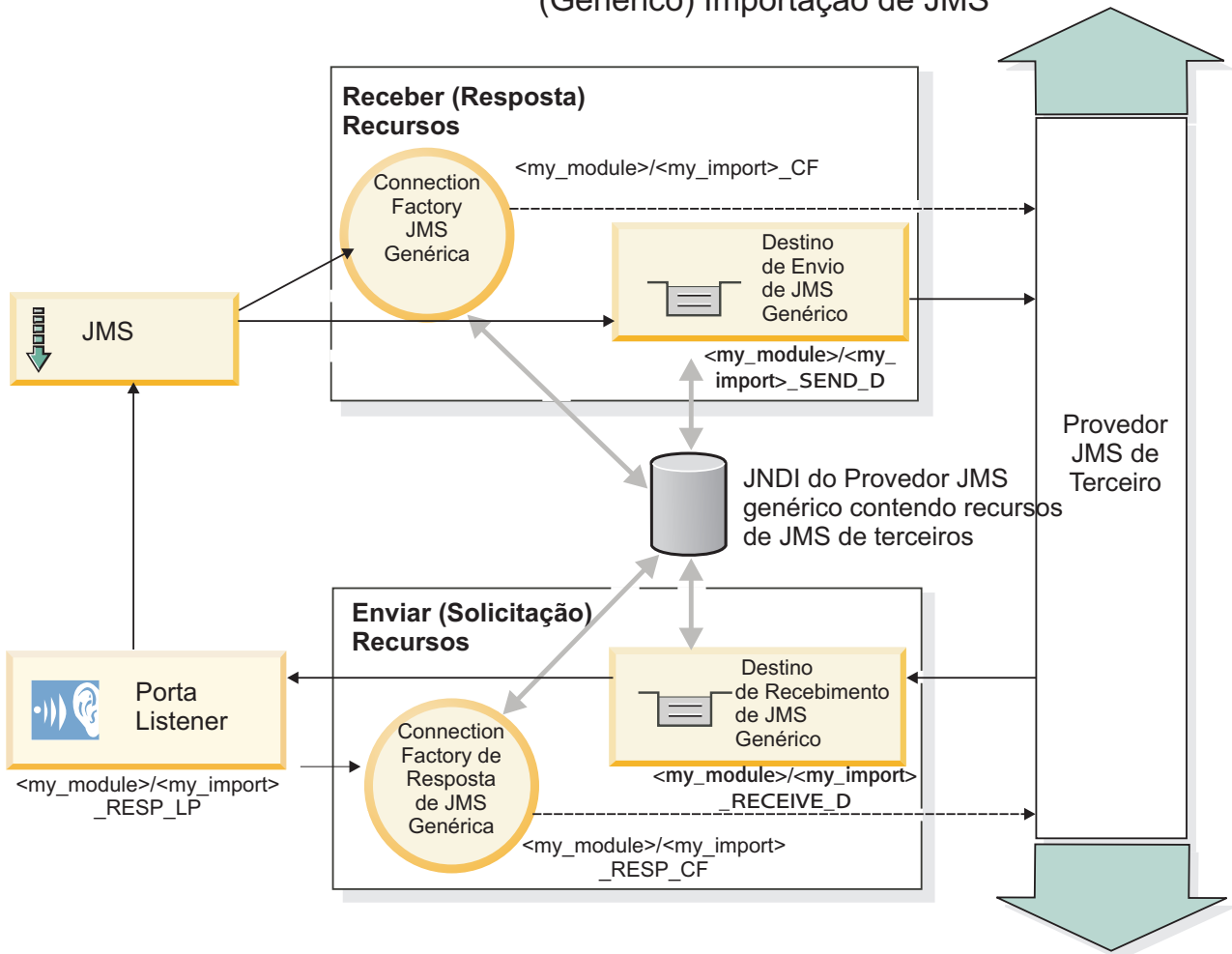


Figura 80. Recursos de Ligação de Importação JMS Genérica

Ligações de Exportação JMS Genéricas

As ligações de exportação JMS genéricas fornecem o meio para módulos SCA fornecerem serviços para aplicativos JMS externos.

A parte da conexão de uma exportação JMS é composta de uma `ConnectionFactory` e uma `ListenerPort`.

Uma exportação do JMS Genérico possui destinos de envio e de recebimento.

- O destino de recebimento é onde a mensagem recebida para o componente de destino deve ser colocada.
- O destino de envio é onde a resposta será enviada, a menos que a mensagem recebida tenha substituído isso usando a propriedade do cabeçalho `replyTo`.

Um MDB é implementado para atender a solicitações recebidas para o destino de recebimento especificado na ligação de exportação.

- O destino especificado no campo de envio será usado para enviar a resposta para a solicitação de entrada, se o componente chamado fornecer uma resposta.
- O destino especificado no campo `replyTo` da mensagem recebida substitui o destino especificado no campo de envio.

- Para cenários de solicitação/resposta, a ligação de importação pode ser configurada (usando o campo **Esquema de Correlação de Resposta** no Integration Designer) para esperar a resposta copiar o ID da mensagem de solicitação no campo ID de Correlação da mensagem de resposta (padrão) ou a resposta pode copiar o ID de correlação no campo ID de Correlação da mensagem de resposta.

O Figura 36 na página 130 ilustra como o solicitante externo está vinculado à exportação.

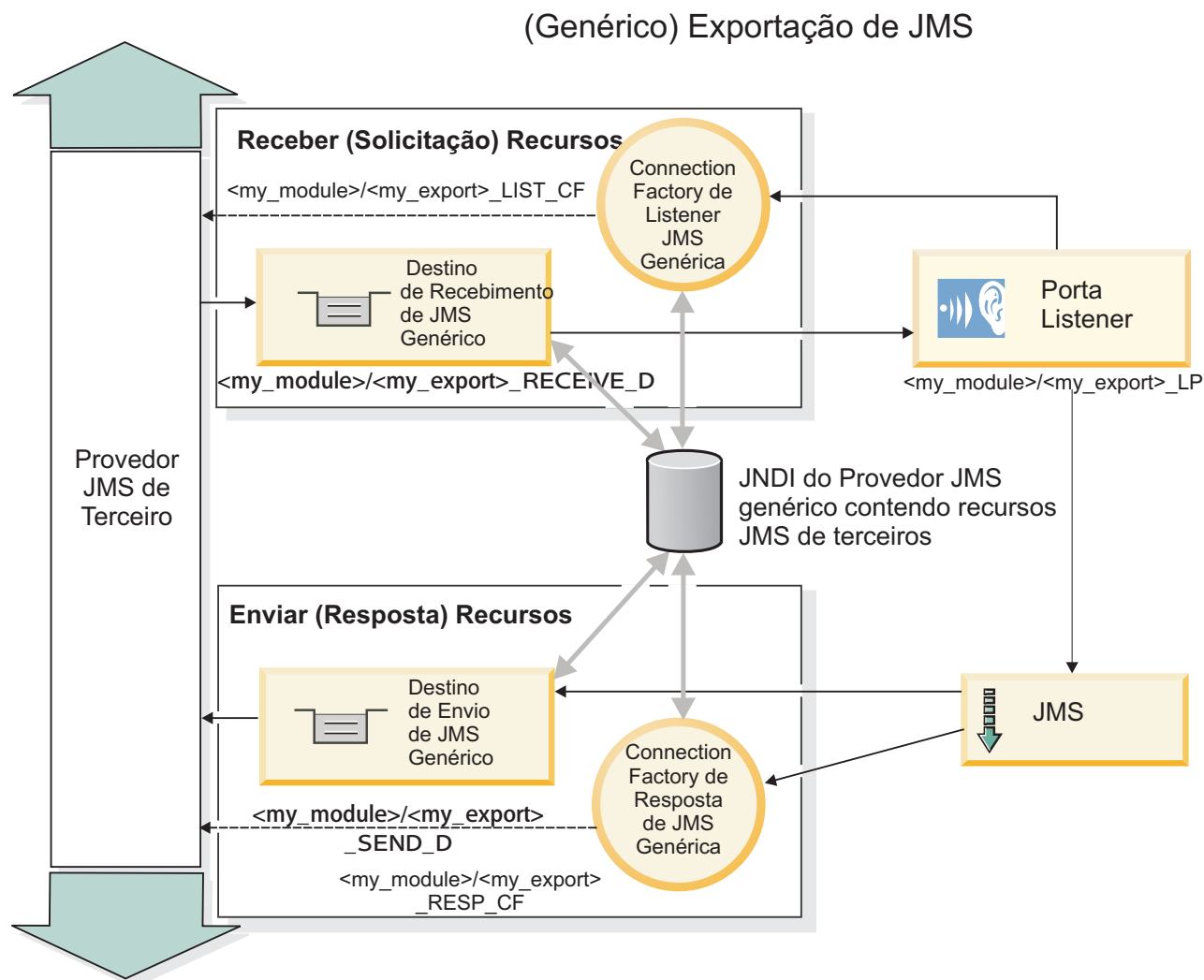


Figura 81. Recursos de Ligação de Exportação JMS Genérica

Recursos-Chave de Ligações JMS Genéricas:

Os recursos da ligação de importação e exportação JMS Genérica são consistentes com aqueles das ligações de importação JMS ou JMS do MQ integradas do WebSphere. Recursos-chave incluem definições de cabeçalho e o acesso a recursos Java EE existentes. No entanto, devido à sua natureza genérica, não há opções de conectividade específicas do provedor JMS e esta ligação tem capacidade limitada para gerar recursos na implementação e instalação.

Importações Genéricas

Como o aplicativo de importação JMS do MQ, a implementação JMS Genérica é assíncrona e suporta três chamadas: unidirecional, bidirecional (também conhecida como solicitação-resposta) e retorno de chamada.

Quando a importação JMS é implementada, um bean acionado por mensagens (MDB) fornecido pelo ambiente de tempo de execução é implementado. O MDB atende as respostas à mensagem de solicitação. O MDB está associado (atende no) ao destino enviado com a solicitação no campo de cabeçalho replyTo da mensagem JMS.

Exportações Genéricas

Ligações de exportação JMS genéricas diferem das ligações de exportação EIS em suas manipulações do retorno do resultado. Uma exportação JMS Genérica envia explicitamente a resposta para o destino replyTo especificado na mensagem recebida. Se nenhum for especificado, o envio de destino será usado.

Quando a exportação JMS Genérica é implementada, um bean acionado por mensagens (um MDB diferente daquele usado para importações JMS Genéricas) é implementado. Ele atende as solicitações recebidas no destino de recebimento e, em seguida, despacha as solicitações a serem processadas pelo tempo de execução SCA.

Cabeçalhos Especiais

Propriedades de cabeçalho especiais são usadas em importações e exportações JMS Genéricas para informar à ligação de destino como manipular a mensagem.

Por exemplo, a propriedade TargetFunctionName é usada pelo seletor de função padrão para identificar o nome da operação na interface de exportação que está sendo chamada.

Nota: A ligação de importação pode ser configurada para definir o cabeçalho TargetFunctionName para o nome da operação para cada operação.

Recursos Java EE

Vários recursos Java EE são criados quando uma ligação JMS é implementada em um ambiente Java EE.

- Porta listener para atender no destino de recebimento (resposta) (apenas bidirecional) para importações e no destino de recebimento (solicitação) para exportações
- Connection factory JMS genérica para outboundConnection (importação) e inboundConnection (exportação)
- Destino JMS genérico para os destinos de envio (importação) e de recebimento (exportação; apenas bidirecional)
- Connection factory JMS genérica para responseConnection (apenas bidirecional e opcional; caso contrário, outboundConnection será usada para importações e inboundConnection é usada para exportações)
- Destino JMS genérico para o destino de recebimento (importação) e de envio (exportação) (apenas bidirecional)
- Destino JMS de retorno de chamada do provedor de sistemas de mensagens padrão usado para acessar o destino de fila de retorno de chamada SIB (apenas bidirecional)
- Connection factory JMS de retorno de chamada de provedor de sistemas de mensagens padrão usada para acessar o destino JMS de retorno de chamada (apenas bidirecional)
- Destino de fila de retorno de chamada SIB usado para armazenar informações sobre a mensagem de solicitação para uso durante o processamento de resposta (apenas bidirecional)

A tarefa de instalação cria a ConnectionFactory, os três destinos e a ActivationSpec a partir das informações nos arquivos de importação e exportação.

Cabeçalhos JMS Genéricos:

Cabeçalhos JMS genéricos são Objetos de Dados de Serviço (SDO) que contêm todas as propriedades das propriedades de mensagem JMS Genéricas. Estas propriedades podem ser da mensagem de entrada ou elas podem ser as propriedades que serão aplicadas na mensagem de saída.

Uma mensagem JMS contém dois tipos de cabeçalhos - o cabeçalho do sistema JMS e diversas propriedades JMS. Ambos os tipos de cabeçalhos podem ser acessados em um módulo de mediação no Objeto de Mensagem de Serviço (SMO) ou usando a API ContextService.

As propriedades a seguir são configuradas estaticamente no methodBinding:

- JMSType
- JMSCorrelationID
- JMSDeliveryMode
- JMSPriority

A ligação JMS Genérica também suporta a modificação dinâmica de cabeçalhos e propriedades JMS da mesma maneira que as ligações JMS e JMS do MQ.

Alguns provedores JMS Genéricos impõem restrições sobre quais propriedades podem ser configuradas pelo aplicativo e em quais combinações. Você deve consultar a documentação do seu produto de terceiros para obter informações adicionais. No entanto, uma propriedade adicional foi incluída no methodBinding, ignoreInvalidOutboundJMSProperties, que permite que quaisquer exceções sejam propagadas.

Os cabeçalhos JMS Genéricos JMS e propriedades de mensagens são usados somente quando o comutador de ligação SCDL da arquitetura do componente de serviço de base está ativado. Quando o comutador está desativado, as informações de contexto são propagadas. Por padrão, esse comutador está ativado. Para evitar a propagação de informações de contexto, altere o valor para **false**.

Quando a propagação de contexto está ativada, as informações de cabeçalho têm permissão para fluir para a mensagem ou para o componente de destino. Para ativar e desativar a propagação de contexto, especifique **true** ou **false** para o atributo contextPropagationEnabled das ligações de importação e de exportação. Por exemplo:

```
<esbBinding xsi:type="eis:JMSImportBinding" contextProagationEnabled="true">
```

O padrão é **true**.

Resolução de Problemas de Ligações JMS Genéricas:

É possível diagnosticar e corrigir problemas com Ligação JMS Genérica.

Exceções de Implementação

Em resposta a várias condições de erro, a implementação de importação e exportação JMS Genérica pode retornar um dos dois tipos de exceções:

- Exceção de Negócios de Serviço: esta exceção é retornada se a falha especificada na interface de negócios de serviço (tipo de porta WSDL) ocorreu.
- Exceção de Tempo de Execução de Serviço: levantada em todos os outros casos. Na maioria dos casos, a exceção de causa conterà a exceção original (JMSException).

Resolução de Problemas de Expiração da Mensagem JMS Genérica

Uma mensagem de pedido pelo provedor JMS está sujeita à expiração.

Expiração de pedido refere-se à expiração de uma mensagem de pedido pelo provedor JMS quando o tempo da `JMSExpiration` na mensagem de pedido é atingido. Como com outras ligações JMS, a ligação JMS Genérica manipula a expiração da solicitação configurando a expiração na mensagem de retorno de chamada posicionada pela importação para ser igual para a solicitação realizada. A notificação da expiração da mensagem de retorno de chamada indicará que a mensagem de pedido expirou e que o cliente deve ser notificado por meio de uma exceção de negócios.

Se o destino do retorno de chamada for movido para um provedor terceirizado, no entanto, esse tipo de expiração de pedido não será suportado.

Expiração de resposta refere-se à expiração de uma mensagem de resposta pelo provedor JMS quando o tempo da `JMSExpiration` na mensagem de resposta é atingido.

A expiração de resposta para a ligação JMS genérica não é suportada, pois o comportamento de expiração exato de um provedor JMS terceirizado não é definido. Entretanto, você pode verificar se a resposta não expirou se e quando ela for recebida.

Para mensagens de pedido de saída, o valor `JMSExpiration` será calculado a partir do tempo aguardado e a partir dos valores `requestExpiration` contidos em `asynchHeader`, se configurados.

Resolução de Problemas de Erros da Connection Factory JMS Genérica

Quando você definir certos tipos de connection factories em seu provedor JMS Genérico, poderá receber uma mensagem de erro quando tentar iniciar um aplicativo. É possível modificar a connection factory externa para evitar este problema.

Ao ativar um aplicativo, você poderá receber a seguinte mensagem de erro:

```
MDB Listener Port JMSConnectionFactory type does not match
JMSDestination type
```

Esse problema pode surgir quando você estiver definindo connection factories externos. Especificamente, a exceção pode ser emitida quando você cria uma Factory de Conexão de Tópico do JMS 1.0.2, em vez de uma Connection Factory do JMS 1.1 (unificado) (ou seja, uma que seja capaz de suportar a comunicação ponto a ponto e de publicação/assinatura).

Para resolver esse problema, execute as seguintes etapas:

1. Acesse o provedor JMS Genérico que você está usando.
2. Substitua o JMS 1.0.2 Topic Connection Factory definido com um JMS 1.1 (unificado) Connection Factory.

Quando ativar o aplicativo com a Connection Factory JMS 1.1 recém-definida, você não deverá mais receber uma mensagem de erro.

Mensagens SCA Baseadas em JMS Genérico Não Estão Aparecendo no Gerenciador de Eventos com Falha

Se mensagens SCA originadas por meio de uma interação do JMS genérico falharem, você poderá encontrar essas mensagens no gerenciador de eventos com falha. Se tais mensagens não aparecerem no gerenciador de eventos com falha, certifique-se de que o valor da propriedade de máximo de novas tentativas na porta listener subjacente seja igual ou maior que 1. A configuração desse valor como 1 ou mais permite a interação com o gerenciador de eventos com falha durante chamadas de SCA para as ligações JMS genéricas.

Exceções de Manipulação:

A maneira na qual a ligação é configurada determina como as exceções que são levantadas por manipuladores de dados ou ligações de dados são manipuladas. Além disso, a natureza do fluxo de mediação determina o comportamento do sistema quando uma exceção desse tipo é lançada.

Uma variedade de problemas pode ocorrer quando um manipulador de dados ou uma ligação de dados é chamada por sua ligação. Por exemplo, um manipulador de dados pode receber uma mensagem que possui uma carga útil corrompida ou pode tentar ler uma mensagem que possua um formato incorreto.

A maneira que sua ligação trata uma exceção desse tipo é determinada por como você implementa o manipulador de dados ou a ligação de dados. O comportamento recomendado é que você projete sua ligação de dados para lançar uma **DataBindingException**.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, qualquer exceção do manipulador de dados é agrupada em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Quando qualquer exceção de tempo de execução, incluindo uma exceção **DataBindingException**, é lançada:

- Se o fluxo de mediação for configurado para ser transacional, a mensagem JMS será armazenada no Gerenciador de Evento com Falha por padrão para exclusão ou reprodução manual.

Nota: É possível alterar o modo de recuperação na ligação de modo que a mensagem seja retrocedida ao invés de ser armazenada no gerenciador de evento com falha.

- Se o fluxo de mediação não for transacional, a exceção será registrada e a mensagem será perdida.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, uma exceção do manipulador de dados é produzida em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Ligações JMS do WebSphere MQ

A ligação JMS do WebSphere MQ fornece integração com aplicativos externos que usam um provedor baseado em JMS do WebSphere MQ.

Use as ligações de exportação e importação de JMS do WebSphere MQ para integrar diretamente com sistemas JMS ou JMS do MQ externos a partir de seu ambiente do servidor. Isso elimina a necessidade de usar os recursos Link do MQ ou Link do Cliente do Barramento de Integração de Serviços.

Quando um componente interage com um serviço baseado em JMS do WebSphere MQ por meio de uma importação, a ligação de importação de JMS do WebSphere MQ utiliza um destino para o qual os dados serão enviados e um destino no qual a resposta pode ser recebida. A conversão dos dados para e a partir de uma mensagem JMS é realizada por meio do componente de extremidade de ligação do manipulador de dados ou dos dados JMS.

Quando um módulo SCA fornece um serviço para clientes JMS do WebSphere MQ, a ligação de exportação de JMS do WebSphere MQ utiliza um destino no qual o pedido pode ser recebido e a resposta pode ser enviada. A conversão dos dados para e a partir de uma mensagem JMS é feita através do manipulador de dados ou da ligação de dados JMS.

O seletor de função fornece um mapeamento para a operação no componente de destino a ser chamado.

Visão Geral de Ligações JMS do WebSphere MQ:

A ligação JMS do WebSphere MQ fornece integração com aplicativos externos que usam o provedor JMS do WebSphere MQ.

Tarefas Administrativas do WebSphere MQ

Espera-se que o administrador do sistema do WebSphere MQ crie o Gerenciador de Filas do WebSphere MQ subjacente, o qual as ligações JMS do WebSphere MQ usarão, antes de executar um aplicativo contendo estas ligações.

Ligações de Importação JMS do WebSphere MQ

A importação JMS do WebSphere MQ permite que os componentes em seu módulo SCA se comuniquem com serviços fornecidos por provedores baseados em JMS do WebSphere MQ. Você deve estar utilizando uma versão suportada do WebSphere MQ. Requisitos detalhados de hardware e software podem ser localizados nas páginas de suporte IBM.

Dois tipos de cenários de uso para ligações de importação JMS do WebSphere MQ são suportados, dependendo do tipo de operação que está sendo chamado:

- **Unidirecional:** A importação JMS do WebSphere MQ coloca uma mensagem no destino de envio configurado na ligação de importação. Nada é enviado para o campo replyTo do cabeçalho JMS.
- **Bidirecional (solicitação-resposta):** A importação JMS do WebSphere MQ coloca uma mensagem no destino de envio.

O destino de recebimento é configurado no campo de cabeçalho replyTo. Um bean acionado por mensagens (MDB) é implementado para atender no destino de recebimento e, quando uma resposta é recebida, o MDB transmite a resposta de volta para o componente.

A ligação de importação pode ser configurada (utilizando o campo **Esquema de Correlação de Resposta** no Integration Designer) para esperar que o ID de correlação da mensagem de resposta tenha sido copiado do ID de mensagem de solicitação (o padrão) ou a partir do ID de correlação da mensagem de solicitação.

Para os cenários de uso unidirecional e bidirecional, propriedades do cabeçalho dinâmicas e estáticas podem ser especificadas. Propriedades estáticas podem ser configuradas a partir da ligação de método da importação JMS. Algumas destas propriedades tem significados especiais para o tempo de execução JMS SCA.

É importante observar que o JMS WebSphere MQ é uma ligação assíncrona. Se um componente de chamada chamar uma importação JMS do WebSphere MQ de maneira síncrona (para uma operação bidirecional), o componente de chamada será bloqueado até que a resposta seja retornada pelo serviço JMS.

Figura 37 na página 136 ilustra como a importação está vinculada ao serviço externo.

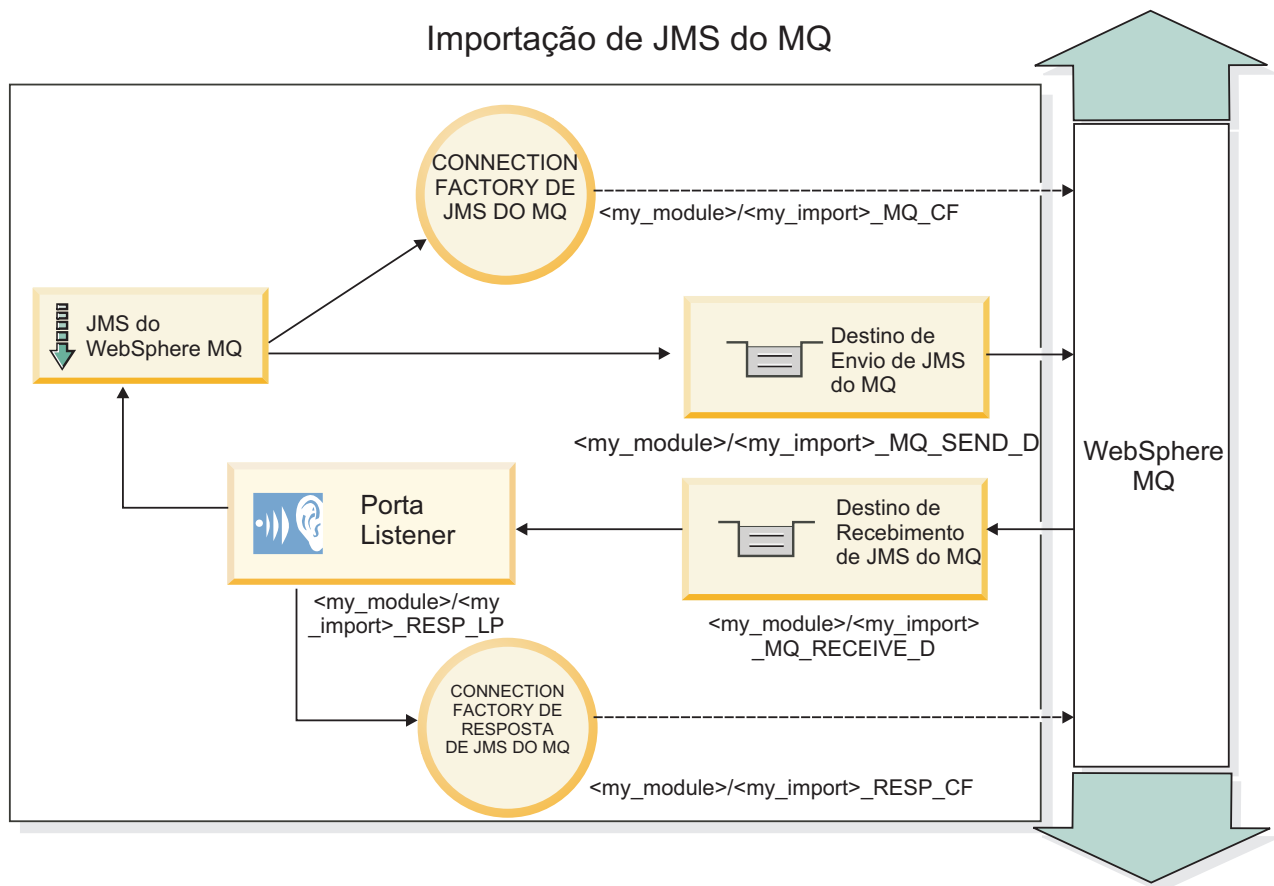


Figura 82. Recursos de Ligação da Importação JMS do WebSphere MQ

Ligações de Exportação JMS do WebSphere MQ

A ligação de exportação JMS do WebSphere MQ fornece os meios para que os módulos SCA forneçam serviços para aplicativos JMS externos no provedor JMS baseado no WebSphere MQ.

Um MDB é implementado para atender a solicitações recebidas para o destino de recebimento especificado na ligação de exportação. O destino especificado no campo Enviar é utilizado para enviar a resposta para a solicitação de entrada se o componente chamado fornecer uma resposta. O destino especificado no campo replyTo da mensagem de resposta substitui o destino especificado no campo Enviar.

O Figura 38 na página 137 ilustra como o solicitante externo está vinculado à exportação.

Exportação de JMS do MQ

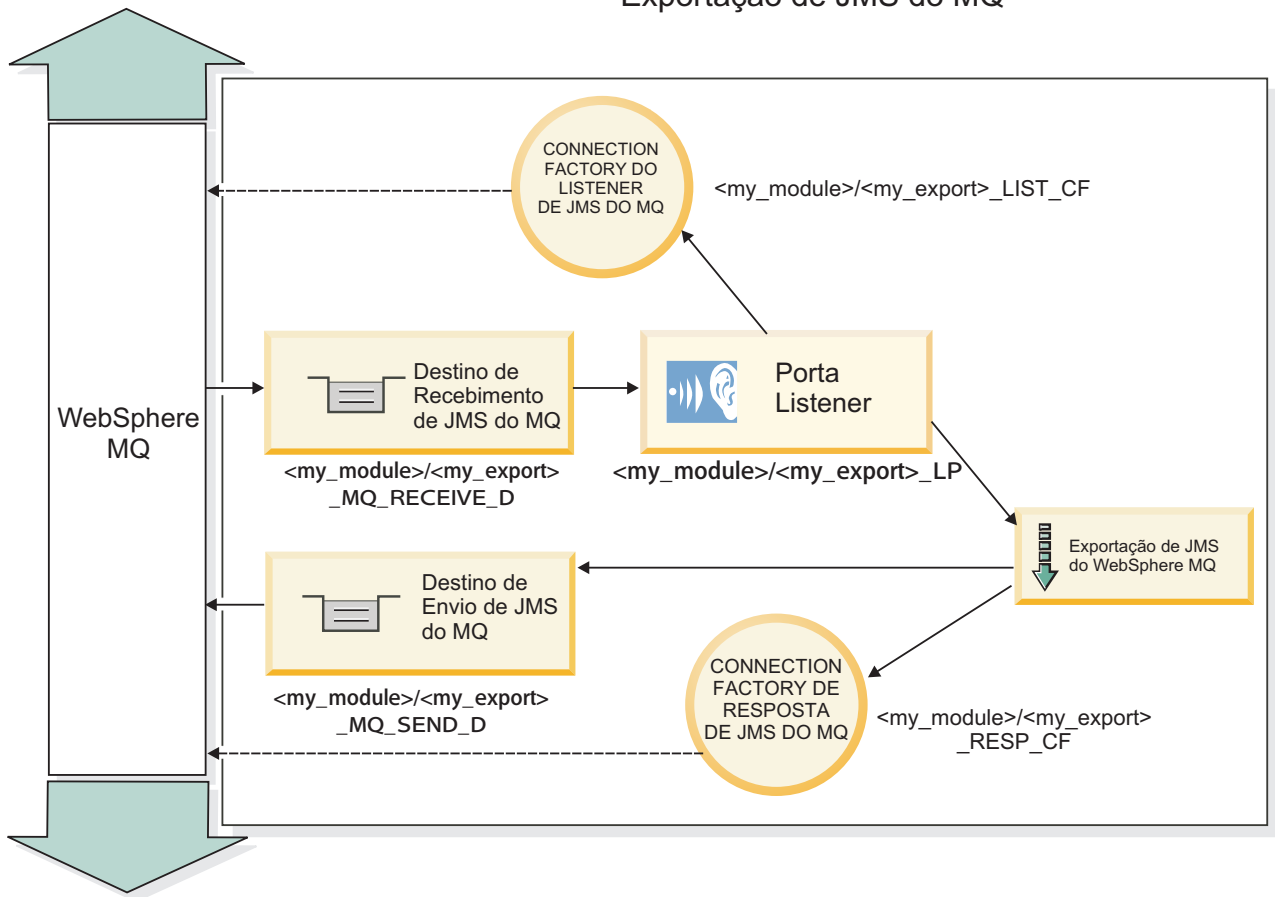


Figura 83. Recursos de Ligação de Exportação JMS do WebSphere MQ

Nota: Figura 37 na página 136 e Figura 38 na página 137 ilustram como um aplicativo de uma versão anterior do IBM Business Process Manager está vinculado a um serviço externo. Para aplicativos desenvolvidos para IBM Business Process Manager Versão 7.0, a Especificação de ativação é usada ao invés da Porta Listener e da Connection Factory.

Recursos-chave das Ligações JMS do WebSphere MQ:

Os recursos-chave das ligações JMS do WebSphere MQ incluem cabeçalhos, artefatos Java EE e recursos Java EE criados.

Cabeçalhos

Um cabeçalho de mensagem JMS contém vários campos predefinidos que contêm valores usados por clientes e provedores para identificar e rotear mensagens. É possível usar propriedades de ligação para configurar esses cabeçalhos com valores fixos ou os cabeçalhos podem ser especificados dinamicamente no tempo de execução.

JMSCorrelationID

Links para uma mensagem relacionada. Geralmente, este campo é configurado com a sequência do identificador de mensagem da mensagem que está sendo respondida.

TargetFunctionName

Este cabeçalho é usado por um dos seletores de função fornecidos para identificar a operação que está sendo chamada. Configurar a propriedade de cabeçalho JMS TargetFunctionName em mensagens enviadas para uma exportação JMS permite que este seletor de função seja usado. A propriedade

pode ser configurada diretamente nos aplicativos clientes JMS ou ao conectar uma importação com uma ligação JMS tal como uma exportação. Neste caso, a ligação de importação JMS deve ser configurada para definir o cabeçalho TargetFunctionName para cada operação na interface com o nome da operação.

Esquemas de Correlação

As ligações JMS do WebSphere MQ fornecem vários esquemas de correlação que são usados para determinar como correlacionar mensagens de solicitação com mensagens de resposta.

RequestMsgIDToCorrelID

O JMSMessageID é copiado no campo JMSCorrelationID. Esta é a configuração padrão.

RequestCorrelIDToCorrelID

O JMSCorrelationID é copiado no campo JMSCorrelationID.

Recursos Java EE

Diversos recursos Java EE são criados quando uma importação JMS do MQ é implementada em um ambiente Java EE.

Parâmetros

Connection Factory do MQ

Usado pelos clientes para criar uma conexão com o provedor JMS do MQ.

Connection Factory de Resposta

Usado pelo tempo de execução JMS do MQ do SCA quando o destino de envio está em um Gerenciador de Filas diferente do destino de recebimento.

Especificação de Ativação

Uma especificação de ativação JMS do MQ está associada a um ou mais beans acionados por mensagens e fornece a configuração necessária para que eles recebam mensagens.

Destinos

- Destino de envio:
 - Importações: Onde a solicitação ou a mensagem de saída é enviada.
 - Exportações: Onde a mensagem de resposta será enviada se não for substituída pelo campo de cabeçalho JMSReplyTo da mensagem recebida.
- Destino de recebimento:
 - Importações: Onde a resposta ou a mensagem recebida deve ser colocada.
 - Exportações: Onde a mensagem recebida ou de solicitação deve ser colocada.

Cabeçalhos JMS:

Uma mensagem JMS contém dois tipos de cabeçalhos - o cabeçalho do sistema JMS e diversas propriedades JMS. Ambos os tipos de cabeçalhos podem ser acessados em um módulo de mediação no Objeto de Mensagem de Serviço (SMO) ou usando a API ContextService.

Cabeçalho do Sistema JMS

O cabeçalho do sistema JMS é representado no SMO pelo elemento JMSHeader, que contém todos os campos geralmente localizados em um cabeçalho JMS. Embora eles possam ser modificados na mediação (ou ContextService), alguns campos de cabeçalho do sistema JMS configurados no SMO não serão propagados na mensagem JMS de saída pois eles são substituídos pelo sistema ou por valores estáticos.

Os campos-chave no cabeçalho do sistema JMS que podem ser atualizados em uma mediação (ou ContextService) são:

- **JMSType** e **JMSCorrelationID** - valores das propriedades de cabeçalho de mensagem predefinidas específicas
- **JMSDeliveryMode** - valores para modo de entrega (persistente ou não persistente; o padrão é persistente)
- **JMSPriority** - valor de prioridade (0 a 9; padrão é JMS_Default_Priority)

Propriedades JMS

As propriedades JMS são representadas no SMO como entradas na lista de Propriedades. As propriedades podem ser incluídas, atualizadas ou excluídas em uma mediação ou usando a API ContextService.

As propriedades também podem ser configuradas estaticamente na ligação JMS. Propriedades que são configuradas estaticamente substituem as configurações (com o mesmo nome) que são configuradas dinamicamente.

As propriedades do usuário propagadas a partir de outras ligações (por exemplo, uma ligação HTTP) serão enviadas na ligação JMS como propriedades JMS.

Configurações de Propagação de Cabeçalho

A propagação do cabeçalho do sistema JMS e das propriedades a partir da mensagem JMS de entrada para componentes de recebimento de dados ou de componentes de envio de dados para a mensagem JMS de saída pode ser controlada pelo sinalizador Propagar Cabeçalho do Protocolo na ligação.

Quando Propagar Cabeçalho de Protocolo é configurado, as informações de cabeçalho têm permitidas para fluírem para a mensagem ou para o componente de destino, conforme descrito na lista a seguir:

- Solicitação de exportação JMS
O cabeçalho JMS recebido na mensagem será propagado para os componentes de destino por meio do serviço de contexto. As propriedades JMS recebidas na mensagem serão propagadas para componentes de destino por meio do serviço de contexto.
- Resposta de exportação JMS
Quaisquer campos de cabeçalho JMS configurados no serviço de contexto serão usados na mensagem de saída, se não substituídos por propriedades estáticas configuradas na ligação de exportação JMS. Quaisquer propriedades configuradas no serviço de contexto serão usadas na mensagem de saída se não substituídas por propriedades estáticas configuradas na ligação de exportação JMS.
- Solicitação de importação JMS
Quaisquer campos de cabeçalho JMS configurados no serviço de contexto serão usados na mensagem de saída, se não substituídos por propriedades estáticas configuradas na ligação de importação JMS. Quaisquer propriedades configuradas no serviço de contexto serão usadas na mensagem de saída se não substituídas por propriedades estáticas configuradas na ligação de importação JMS.
- Resposta de importação JMS
O cabeçalho JMS recebido na mensagem será propagado para os componentes de destino por meio do serviço de contexto. As propriedades JMS recebidas na mensagem serão propagadas para componentes de destino por meio do serviço de contexto.

Clientes Externos:

O servidor pode enviar mensagens para clientes externos ou receber mensagens deles usando ligações JMS do WebSphere MQ.

Um cliente externo (como um portal da web ou um Enterprise Information System) pode enviar uma mensagem para um componente SCA no aplicativo por meio de uma exportação ou pode ser chamado por um componente SCA no aplicativo por meio de uma importação.

A ligação de exportação JMS do WebSphere MQ implementa beans conduzidos por mensagens (MDBs) para atender as solicitações recebidas no destino de recebimento especificado na ligação de exportação. O destino especificado no campo de envio é usado para enviar a resposta para a solicitação de entrada se o aplicativo chamado fornece uma resposta. Assim, um cliente externo pode chamar aplicativos por meio da ligação de exportação.

As importações JMS do WebSphere MQ se ligam a clientes externos e podem entregar mensagem para eles. Esta mensagem pode ou não demandar uma resposta do cliente externo.

Mais informações sobre como interagir com clientes externos usando o WebSphere MQ podem ser localizadas no Centro de Informações do WebSphere MQ.

Resolução de Problemas de Ligações JMS do WebSphere MQ:

É possível diagnosticar e corrigir problemas com as ligações JMS do WebSphere MQ.

Exceções de Implementação

Em resposta a várias condições de erro, a implementação de importação e exportação de JMS do MQ pode retornar um de dois tipos de exceções:

- Exceção de Negócios de Serviço: esta exceção é retornada se a falha especificada na interface de negócios de serviço (tipo de porta WSDL) ocorreu.
- Exceção de Tempo de Execução de Serviço: levantada em todos os outros casos. Na maioria dos casos, a exceção de causa conterá a exceção original (JMSEException).

Por exemplo, uma importação espera apenas uma mensagem de resposta para cada mensagem de solicitação. Se mais de uma resposta chegar, ou se uma resposta atrasada (uma para a qual a expiração de resposta de SCA expirou) chegar, uma Exceção de Tempo de Execução de Serviço será emitida. A transação é retrocedida e a mensagem de resposta é retirada da fila ou manipulada pelo gerenciador de eventos com falha.

Mensagens SCA com Base no WebSphere MQ JMS Não Aparecem no Gerenciador de Eventos com Falha

Se mensagens SCA originadas por uma interação do WebSphere MQ JMS falharem, você pode encontrar essas mensagens no gerenciador de eventos com falha. Se tais mensagens não estiverem aparecendo no gerenciador de eventos com falha, certifique-se de que o valor da propriedade de máximo de novas tentativas na porta listener subjacente seja igual ou maior que **1**. A configuração deste valor como **1** ou mais permite a interação com o gerenciador de eventos com falha durante chamadas de SCA para as ligações JMS do MQ.

Cenários de Uso Incorreto: Comparação com Ligações WebSphere MQ

A ligação WebSphere MQ JMS é projetada para interoperar com aplicativos JMS implementados no WebSphere MQ, que expõe mensagens de acordo com o modelo de mensagem JMS. A importação e a exportação do WebSphere MQ, no entanto, são projetadas principalmente para interoperar com aplicativos WebSphere MQ nativos e expor o conteúdo integral do corpo da mensagem do WebSphere MQ para mediações.

Os cenários a seguir devem ser construídos utilizando a ligação do WebSphere MQ JMS, não a ligação do WebSphere MQ:

- Chamar um MDB (bean acionado por mensagens) JMS a partir de um módulo SCA, no qual o MDB é implementado no provedor WebSphere MQ JMS. Use uma importação JMS do WebSphere MQ.
- Permitir que o módulo SCA seja chamado a partir de um servlet de componente Java EE ou EJB por meio de JMS. Use uma exportação JMS do WebSphere MQ.

- Mediar o conteúdo de uma JMS MapMessage, em trânsito através do WebSphere MQ. Use uma exportação e importação JMS do WebSphere MQ em conjunto com o manipulador de dados ou a ligação de dados apropriada.

Há situações em que a ligação WebSphere MQ e a ligação WebSphere MQ JMS talvez sejam esperadas para interoperação. Em particular, quando fizer uma ponte entre os aplicativos WebSphere MQ Java EE e não Java EE, use uma exportação do WebSphere MQ e importação JMS do WebSphere MQ (ou vice-versa) em conjunto com ligações de dados ou módulos de mediação apropriados (ou ambos).

Exceções de Manipulação:

A maneira na qual a ligação é configurada determina como as exceções que são levantadas por manipuladores de dados ou ligações de dados são manipuladas. Além disso, a natureza do fluxo de mediação determina o comportamento do sistema quando uma exceção desse tipo é lançada.

Uma variedade de problemas pode ocorrer quando um manipulador de dados ou uma ligação de dados é chamada por sua ligação. Por exemplo, um manipulador de dados pode receber uma mensagem que possui uma carga útil corrompida ou pode tentar ler uma mensagem que possua um formato incorreto.

A maneira que sua ligação trata uma exceção desse tipo é determinada por como você implementa o manipulador de dados ou a ligação de dados. O comportamento recomendado é que você projete sua ligação de dados para lançar uma **DataBindingException**.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, qualquer exceção do manipulador de dados é agrupada em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Quando qualquer exceção de tempo de execução, incluindo uma exceção **DataBindingException**, é lançada:

- Se o fluxo de mediação for configurado para ser transacional, a mensagem JMS será armazenada no Gerenciador de Evento com Falha por padrão para exclusão ou reprodução manual.

Nota: É possível alterar o modo de recuperação na ligação de modo que a mensagem seja retrocedida ao invés de ser armazenada no gerenciador de evento com falha.

- Se o fluxo de mediação não for transacional, a exceção será registrada e a mensagem será perdida.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, uma exceção do manipulador de dados é produzida em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Ligações WebSphere MQ

A ligação do WebSphere MQ fornece conectividadeService Component Architecture (SCA) com os aplicativos do WebSphere MQ.

Use as ligações de importação e exportação do WebSphere MQ para integrar diretamente a um sistema baseado no WebSphere MQ a partir do ambiente do servidor. Isso elimina a necessidade de usar os recursos Link do MQ ou Link do Cliente do Barramento de Integração de Serviços.

Quando um componente interagir com um serviço do WebSphere MQ por meio de uma importação, a ligação de importação do WebSphere MQ usará uma fila para a qual os dados serão enviados e uma fila em que a resposta possa ser recebida.

Quando um módulo SCA fornecer um serviço para os clientes do WebSphere MQ, a ligação de exportação do WebSphere MQ usará uma fila em que a solicitação poderá ser recebida e a resposta poderá ser enviada. O seletor de função fornece um mapeamento para a operação no componente de destino a ser chamado.

A conversão dos dados de carga útil para/de uma mensagem do MQ é feita por meio do manipulador de dados do corpo do MQ ou da ligação de dados. A conversão dos dados do cabeçalho para/de uma mensagem do MQ é feita por meio da ligação de dados do cabeçalho do MQ.

Para obter informações sobre as versões do WebSphere MQ suportadas, consulte a página da web Requisitos Detalhados do Sistema.

Visão Geral de Ligações do WebSphere MQ:

A ligação do WebSphere MQ fornece integração com aplicativos baseados em MQ nativos.

Tarefas Administrativas do WebSphere MQ

Espera-se que o administrador do sistema do WebSphere MQ crie o WebSphere MQ Queue Manager subjacente, que as ligações do WebSphere MQ irão usar, antes de executar um aplicativo contendo estas ligações.

Tarefas Administrativas do WebSphere

Você deve configurar a propriedade **Caminho da biblioteca nativa** do adaptador de recursos do MQ no Websphere para a versão do WebSphere MQ suportada pelo servidor e reiniciar o servidor. Isto garante que as bibliotecas de uma versão suportada do WebSphere MQ estejam sendo usadas. Os requisitos de hardware e software detalhados podem ser localizados nas páginas de suporte IBM.

Ligações de Importação do WebSphere MQ

A ligação de importação do WebSphere MQ permite que componentes em seu módulo SCA se comuniquem com serviços fornecidos por aplicativos externos baseados em WebSphere MQ. Você deve estar utilizando uma versão suportada do WebSphere MQ. Os requisitos de hardware e software detalhados podem ser localizados nas páginas de suporte IBM.

A interação com sistemas WebSphere MQ externos inclui o uso de filas para o envio de solicitações e recebimento de respostas.

Dois tipos de cenários de uso para a ligação de importação do WebSphere MQ são suportados, dependendo do tipo de operação que está sendo chamado:

- Unidirecional: A importação do WebSphere MQ coloca uma mensagem na fila configurada no campo **Fila de Destino de Envio** da ligação de importação. Nada é enviado para o campo replyTo do cabeçalho MQMD.
- Bidirecional (solicitação-resposta): A importação do WebSphere MQ coloca uma mensagem na fila configurada no campo **Fila de Destino de Envio**

A fila de recebimento é configurada no campo replyTo do cabeçalho MQMD. Um bean acionado por mensagens (MDB) é implementado para atender na fila de recebimento, e quando uma resposta é recebida, o MDB passa a resposta de volta para o componente.

A ligação de importação pode ser configurada (usando o campo **Esquema de Correlação de Resposta**) para esperar que o ID de correlação da mensagem de resposta tenha sido copiado do ID de mensagem de solicitação (o padrão) ou do ID de correlação da mensagem de solicitação.

É importante observar que o WebSphere MQ é uma ligação assíncrona. Se um componente de chamada chamar uma importação do WebSphere MQ de maneira síncrona (para uma operação bidirecional), o componente de chamada será bloqueado até a resposta ser retornada pelo serviço do WebSphere MQ.

Figura 39 na página 143 ilustra como a importação está vinculada ao serviço externo.

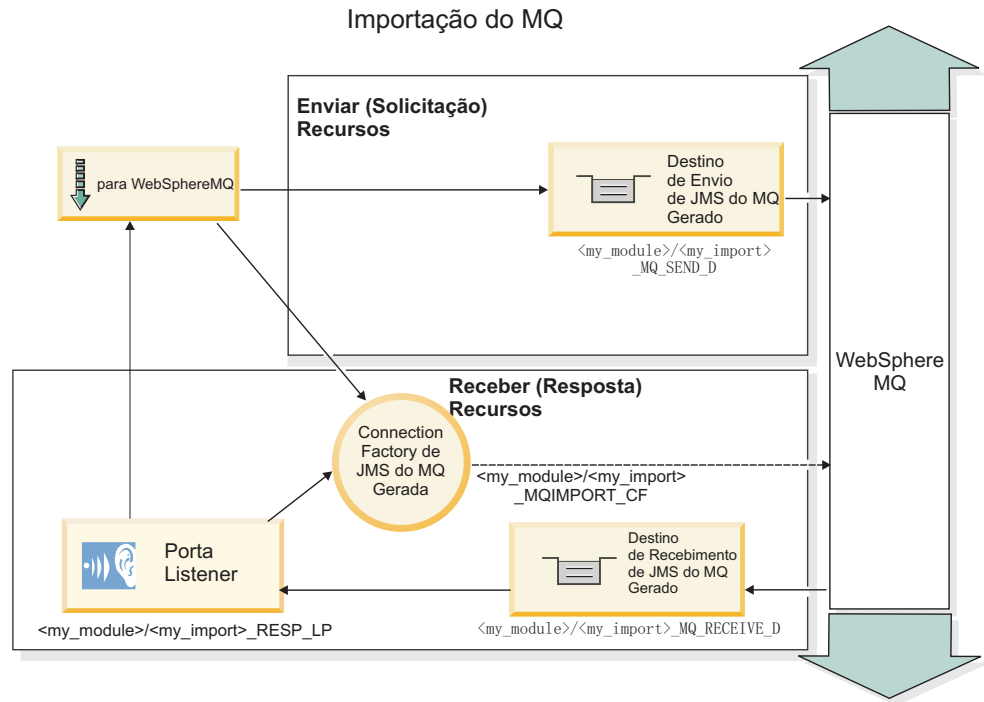


Figura 84. Recursos de Ligação de Importação do WebSphere MQ

Ligações de Exportação do WebSphere MQ

A ligação de exportação do WebSphere MQ fornece os meios para que os módulos SCA forneçam serviços para aplicativos externos baseados no WebSphere MQ.

Um MDB é implementado para atender a solicitações recebidas para a **Fila de destino de recebimento** especificada na ligação de exportação. A fila especificada no campo **Fila de Destino de Envio** é usada para enviar a resposta para a solicitação de entrada se o componente chamado fornecer uma resposta. A fila especificada no campo replyTo da mensagem de resposta substitui a fila especificada no campo **Fila de Destino de Envio**.

O Figura 40 na página 144 ilustra como o solicitante externo está vinculado à exportação.

Exportação do MQ

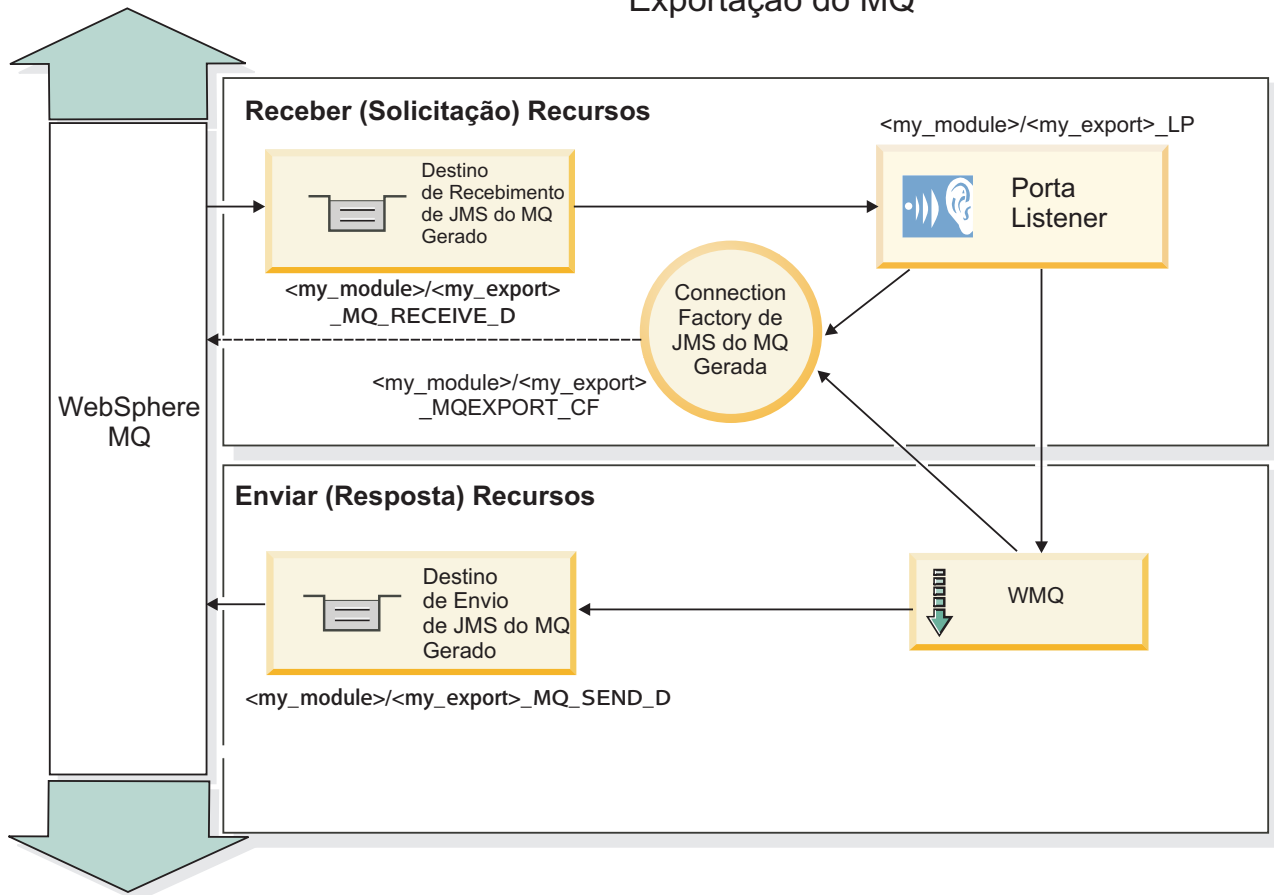


Figura 85. Recursos de Ligação de Exportação do WebSphere MQ

Nota: Figura 39 na página 143 e Figura 40 na página 144 ilustram como um aplicativo de uma versão anterior do IBM Business Process Manager está vinculado a um serviço externo. Para aplicativos desenvolvidos para o IBM Business Process Manager Versão 7.x ou posterior, a especificação de ativação é usada, em vez da Porta do Listener e do Connection Factory.

Recursos-chave de uma Ligação do WebSphere MQ:

Os recursos-chave de uma ligação do WebSphere MQ incluem cabeçalhos, artefatos Java EE e recursos Java EE criados.

Esquemas de Correlação

Um aplicativo de solicitação/resposta do WebSphere MQ pode usar uma de várias técnicas para correlacionar as mensagens de resposta com solicitações, construídas em torno dos campos MQMD MessageID e CorrelID. Na grande maioria dos casos, o solicitante permite que o gerenciador de filas selecione um MessageID e espera que o aplicativo de resposta copie isso para o CorrelID da resposta. Na maioria dos casos, o solicitante e o aplicativo de resposta sabem, implicitamente, qual técnica de correlação está em uso. Ocasionalmente, o aplicativo de resposta honrará vários sinalizadores no campo Relatório da solicitação que descrevem como lidar com esses campos.

As ligações de exportação para o WebSphere MQ podem ser configuradas com as seguintes opções:

Opções MsgId de resposta:

Novo MsgID

Permite que o gerenciador de filas selecione um MsgId exclusivo para a resposta (padrão).

Copiar a partir do MsgID de Solicitação

Copia o campo MsgId a partir do campo MsgId na solicitação.

Copiar a partir da mensagem SCA

Configura o MsgId para que seja executado nos cabeçalhos do WebSphere MQ na mensagem de resposta SCA ou permite que o gerenciador de filas defina um novo ID se o valor não existir.

Como Opções de Relatório

Inspeciona o campo Relatório do MQMD na solicitação para obter uma dica de como manipular o MsgId. As opções MQRO_NEW_MSG_ID e MQRO_PASS_MSG_ID são suportadas e se comportam como Novo MsgId e Copiar de MsgID de Solicitação.

Opções de CorrelId de resposta:**Copiar a partir do MsgID de Solicitação**

Copia o campo CorrelId a partir do campo MsgId na solicitação (padrão).

Copiar a partir do CorrelID de Solicitação

Copia o campo CorrelId a partir do campo CorrelId na solicitação.

Copiar a partir da mensagem SCA

Configura o CorrelId para ser executado nos cabeçalhos do WebSphere MQ na mensagem de resposta SCA ou o deixa em branco se o valor não existir.

Como Opções de Relatório

Inspeciona o campo Relatório do MQMD na solicitação para obter uma dica de como manipular o CorrelId. As opções MQRO_COPY_MSG_ID_TO_CORREL_ID e MQRO_PASS_CORREL_ID são suportadas e se comportam como Copiar de MsgID de Solicitação e Copiar de CorrelID de Solicitação.

As ligações de importação para mensagens do WebSphere MQ podem ser configuradas com as opções a seguir:

Opções MsgId de solicitação:**Novo MsgID**

Permite que o gerenciador de filas selecione um MsgId exclusivo para a solicitação (padrão).

Copiar a partir da mensagem SCA

Configura o MsgId a ser executado nos cabeçalhos do WebSphere MQ na mensagem de solicitação SCA ou permite que o gerenciador de filas defina um novo Id se o valor não existir.

Opções de correlação de resposta:**Resposta possui CorrelID copiado de MsgId**

Espera que a mensagem de resposta tenha o campo CorrelId configurado, de acordo com o MsgId da solicitação (padrão).

Resposta possui MsgID copiado de MsgId

Espera que a mensagem de resposta tenha um campo MsgId configurado, de acordo com o MsgId da solicitação.

Resposta possui CorrelID copiado de CorrelId

Espera que a mensagem de resposta tenha um campo CorrelId configurado, de acordo com o CorrelId da solicitação.

Recursos Java EE

Vários recursos Java EE são criados quando uma ligação do WebSphere MQ é implementada em um ambiente Java EE.

Parâmetros

Connection Factory do MQ

Usado pelos clientes para criar uma conexão com o provedor do WebSphere MQ.

Connection Factory de Resposta

Usado pelo tempo de execução do MQ do SCA quando o destino de envio está em um Gerenciador de Filas diferente do destino de recebimento.

Especificação de Ativação

Uma especificação de ativação JMS do MQ está associada a um ou mais beans acionados por mensagens e fornece a configuração necessária para que eles recebam mensagens.

Destinos

- Destino de envio: onde a mensagem de solicitação ou de saída é enviada (importação); onde a mensagem de resposta será enviada (exportação), se não substituída pelo campo de cabeçalho do MQMD ReplyTo na mensagem recebida.
- Destino de recebimento: onde a mensagem de resposta/solicitação ou recebida deve ser colocada.

Cabeçalhos do WebSphere MQ:

Os cabeçalhos do WebSphere MQ incorporam determinadas convenções para conversão para as mensagens de Service Component Architecture (SCA).

As mensagens do WebSphere MQ consistem em um cabeçalho do sistema (o MQMD), zero ou mais outros cabeçalhos MQ (sistema ou customizado) e um corpo da mensagem. Se diversos cabeçalhos da mensagem existirem na mensagem, a ordem dos cabeçalhos será importante.

Cada cabeçalho contém informações que descrevem a estrutura do cabeçalho a seguir. O MQMD descreve o primeiro cabeçalho.

Como Cabeçalhos MQ são Analisados

Uma ligação de dados do Cabeçalho MQ é usada para analisar os cabeçalhos MQ. Os seguintes cabeçalhos são suportados automaticamente:

- MQRFH
- MQRFH2
- MQCIH
- MQIIH

Cabeçalhos que iniciam com **MQH** são manipulados de maneira diferente. Campos específicos do cabeçalho não são analisados; eles permanecem como bytes não analisados.

Para outros cabeçalhos do MQ, é possível gravar ligações de dados do cabeçalho MQ customizadas para analisar esses cabeçalhos.

Como Cabeçalhos MQ são Acessados

Os cabeçalhos MQ podem ser acessados no produto de uma de duas maneiras:

- Por meio do objeto de mensagem de serviço (SMO) em uma mediação
- Por meio da API ContextService

Cabeçalhos MQ são representados internamente com o elemento SMO MQHeader. MQHeader é um contêiner de dados de cabeçalho que estende MQControl, mas contém um elemento de valor de anyType. Ele contém o MQMD, o MQControl (informações de controle do corpo da mensagem MQ) e uma lista de outros cabeçalhos MQ.

- MQMD representa o conteúdo da descrição de mensagem do WebSphere MQ, exceto para informações que determinam a estrutura e a codificação do corpo.
- MQControl contém informações que determinam a estrutura e a codificação de um corpo da mensagem.
- MQHeaders contém uma lista de objetos MQHeader.

A cadeia do cabeçalho MQ é verificada para que, dentro do SMO, cada cabeçalho MQ transporte suas próprias informações de controle (CCSID, Codificação e Formato). Os cabeçalhos podem ser incluídos ou excluídos facilmente, sem alterar os outros dados do cabeçalho.

Configurando Campos no MQMD

É possível atualizar o MQMD usando a API de Contexto ou através do objeto de mensagem de serviço (SMO) em uma mediação. Os campos a seguir são propagados automaticamente para a mensagem do MQ de saída:

- Codificação
- Conjunto de Caracteres Codificados
- Formato
- Relatório
- Expiração
- Feedback
- Prioridade
- Persistência
- ID de Correlação
- Sinalizadores de Msg

Configure a ligação do MQ em uma Importação ou Exportação para propagar as propriedades a seguir para a mensagem MQ de saída:

MsgID

Configure **ID da Mensagem de Solicitação** como cópia da mensagem SCA.

MsgType

Limpe a caixa de seleção **Configurar tipo de mensagem como MQMT_DATAGRAM ou MQMT_REQUEST para operação de solicitação-resposta.**

ReplyToQ

Limpe a caixa de seleção **Substituir resposta na fila da mensagem de solicitação.**

ReplyToQMGr

Limpe a caixa de seleção **Substituir resposta na fila da mensagem de solicitação.**

Da versão 7.0 em diante, campos de contexto podem ser substituídos usando uma propriedade customizada na definição de destino de JNDI. Configure a propriedade customizada MDCTX com o valor SET_IDENTITY_CONTEXT no destino de envio para propagar os campos a seguir para a mensagem do MQ de saída:

- UserIdentifier
- AppIdentityData

Configure a propriedade customizada MDCTX com o valor SET_ALL_CONTEXT no destino de envio para propagar as propriedades a seguir para a mensagem do MQ de saída:

- UserIdentifier
- AppIdentityData
- PutApplType
- PutApplName
- ApplOriginData

Alguns campos não são propagados para a mensagem do MQ de saída. Os campos a seguir são substituídos durante o envio da mensagem:

- BackoutCount
- AccountingToken
- PutDate
- PutTime
- Deslocamento
- OriginalLength

Incluindo o MQCIH estaticamente em uma ligação do WebSphere MQ:

O IBM Business Process Manager suporta a inclusão de informações de cabeçalho MQCIH estaticamente sem usar um módulo de mediação.

Há várias maneiras de incluir as informações de cabeçalho MQCIH em uma mensagem (por exemplo, usando a primitiva de mediação do Configurador de Cabeçalho). Pode ser útil incluir estas informações de cabeçalho estaticamente, sem o uso de um módulo de mediação adicional. As informações de cabeçalho estático, incluindo o nome do programa CICS, o ID de transação e outros detalhes do cabeçalho de formato de dados, podem ser definidas e incluídas como parte da ligação do WebSphere MQ.

O WebSphere MQ, o MQ CICS Bridge e o CICS devem ser configurados para informações de cabeçalho MQCIH para serem incluídos estaticamente.

É possível usar Integration Designer para configurar a importação do WebSphere MQ com os valores estáticos que são necessários para as informações de cabeçalho MQCIH.

Quando uma mensagem chega e é processada pela importação do WebSphere MQ, é feita uma verificação para ver se as informações do cabeçalho MQCIH já estão presentes na mensagem. Se o MQCIH estiver presente, os valores estáticos definidos na importação do WebSphere MQ serão usados para substituir os valores dinâmicos correspondentes na mensagem. Se o MQCIH não estiver presente, um será criado na mensagem e os valores estáticos definidos na importação do WebSphere MQ serão incluídos.

Os valores estáticos definidos na importação do WebSphere MQ são específicos para um método. É possível especificar diferentes valores de MQCIH estáticos para métodos diferentes dentro da mesma importação do WebSphere MQ.

Este recurso não é usado para fornecer valores padrão se o MQCIH não contém informações do cabeçalho específicas porque um valor estático definido na importação do WebSphere MQ substituirá um valor correspondente fornecido na mensagem recebida.

Cientes Externos:

O IBM Business Process Manager pode enviar mensagens aos clientes externos e receber mensagens deles usando ligações do WebSphere MQ.

Um cliente externo (por exemplo, um portal da web ou um Enterprise Information System) pode enviar uma mensagem a um componente SCA no aplicativo por meio de uma exportação ou pode ser chamado por um componente SCA no aplicativo por meio de uma importação.

A ligação de exportação do WebSphere MQ implementa beans conduzidos por mensagens (MDBs) para atender as solicitações recebidas no destino de recebimento especificado na ligação de exportação. O destino especificado no campo de envio é usado para enviar a resposta para a solicitação de entrada se o aplicativo chamado fornece uma resposta. Assim, um cliente externo pode chamar aplicativos por meio da ligação de exportação.

As importações do WebSphere MQ se ligam aos clientes externos e podem entregar mensagem para eles. Esta mensagem pode ou não demandar uma resposta do cliente externo.

Mais informações sobre como interagir com clientes externos usando o WebSphere MQ podem ser localizadas no Centro de Informações do WebSphere MQ.

Resolução de Problemas de Ligações WebSphere MQ:

Você pode diagnosticar e corrigir falhas e condições de falhas que ocorrem com ligações WebSphere MQ.

Principais Condições de Falha

As principais condições de falha das ligações WebSphere MQ são determinadas pelas semântica transacional, pela configuração do WebSphere MQ ou pela referência ao comportamento existente em outros componentes. As principais condições de falha incluem:

- Falha de conexão com o gerenciador de fila ou a fila do WebSphere MQ.
Uma falha de conexão com o WebSphere MQ para receber mensagens resultará na Porta Listener MDB falhando ao iniciar. Essa condição será registrada no log do WebSphere Application Server. As mensagens persistentes permanecerão na fila do WebSphere MQ até que elas sejam recuperadas com sucesso (ou expiradas pelo WebSphere MQ).
Uma falha de conexão com o WebSphere MQ para enviar mensagens de saída causará o retrocesso da transação que controla o envio.
- Falha ao analisar uma mensagem de entrada ou ao construir uma mensagem de saída.
Uma falha na ligação de dados causa o retrocesso da transação que controla o trabalho.
- Falha ao enviar a mensagem de saída.
Uma falha ao enviar uma mensagem causa o retrocesso da transação relevante.
- Mensagens de resposta diversas ou inesperadas.
A importação espera apenas uma mensagem de resposta para cada mensagem de solicitação. Se mais de uma resposta chegar, ou se uma resposta atrasada (uma para a qual a expiração de resposta de SCA expirou) chegar, uma Exceção de Tempo de Execução de Serviço será emitida. A transação é retrocedida e a mensagem de resposta é retirada da fila ou manipulada pelo gerenciador de eventos com falha.

Cenários de Uso Incorreto: Comparação com Ligações WebSphere MQ JMS

A importação e exportação do WebSphere MQ são projetadas principalmente para interoperar com aplicativos nativos do WebSphere MQ e exportar o conteúdo completo do corpo da mensagem do

WebSphere MQ para mediações. A ligação WebSphere MQ JMS, entretanto, é projetada para interoperar com aplicativos JMS implementados no WebSphere MQ, que expõe mensagens de acordo com o modelo de mensagem JMS.

Os cenários a seguir devem ser construídos utilizando a ligação do WebSphere MQ JMS, não a ligação do WebSphere MQ:

- Chamar um MDB (bean acionado por mensagens) JMS a partir de um módulo SCA, no qual o MDB é implementado no provedor WebSphere MQ JMS. Use uma importação JMS do WebSphere MQ.
- Permitir que o módulo SCA seja chamado a partir de um servlet de componente Java EE ou EJB por meio de JMS. Use uma exportação JMS do WebSphere MQ.
- Mediar o conteúdo de uma JMS MapMessage, em trânsito através do WebSphere MQ. Use uma exportação e uma importação JMS do WebSphere MQ em conjunto com a ligação de dados apropriada.

Há situações em que a ligação WebSphere MQ e a ligação WebSphere MQ JMS talvez sejam esperadas para interoperação. Em particular, quando fizer uma ponte entre os aplicativos WebSphere MQ Java EE e não Java EE, use uma exportação do WebSphere MQ e importação JMS do WebSphere MQ (ou vice-versa) em conjunto com ligações de dados ou módulos de mediação apropriados (ou ambos).

Mensagens Não Entregues

Se o WebSphere MQ não puder entregar uma mensagem para seu destino desejado (devido a erros de configuração, por exemplo), ele enviará as mensagens para uma fila de devoluções nomeada.

Ao fazer isso, ele inclui um cabeçalho de devoluções no início do corpo da mensagem. Este cabeçalho contém os motivos da falha, o destino original e outras informações.

Mensagens SCA Baseadas em MQ Não Aparecerem no Gerenciador de Eventos com Falha

Se mensagens SCA se originarem devido a uma falha de interação do WebSphere MQ, você poderá encontrar essas mensagens no gerenciador de eventos com falha. Se estas mensagens não estão aparecendo no gerenciador de eventos com falha, verifique se o destino do WebSphere MQ subjacente tem um valor máximo de entregas com falha maior do que 1. A configuração deste valor como 2 ou mais permite a interação com o gerenciador de eventos com falha durante chamadas de SCA para ligações do WebSphere MQ.

Eventos com Falha do MQ são Reproduzidas no Gerenciador de Filas Errado

Quando uma connection factory predefinida deve ser usada para conexões de saída, as propriedades de conexão devem corresponder àquelas definidas na especificação de ativação usada para conexões de entrada.

A connection factory predefinida é usada para criar uma conexão ao reproduzir um evento com falha e deve, portanto, ser configurada para usar o mesmo gerenciador de filas a partir do qual a mensagem foi recebida originalmente.

Exceções de Manipulação:

A maneira na qual a ligação é configurada determina como as exceções que são levantadas por manipuladores de dados ou ligações de dados são manipuladas. Além disso, a natureza do fluxo de mediação determina o comportamento do sistema quando uma exceção desse tipo é lançada.

Uma variedade de problemas pode ocorrer quando um manipulador de dados ou uma ligação de dados é chamada por sua ligação. Por exemplo, um manipulador de dados pode receber uma mensagem que possui uma carga útil corrompida ou pode tentar ler uma mensagem que possua um formato incorreto.

A maneira que sua ligação trata uma exceção desse tipo é determinada por como você implementa o manipulador de dados ou a ligação de dados. O comportamento recomendado é que você projete sua ligação de dados para lançar uma **DataBindingException**.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, qualquer exceção do manipulador de dados é agrupada em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Quando qualquer exceção de tempo de execução, incluindo uma exceção **DataBindingException**, é lançada:

- Se o fluxo de mediação for configurado para ser transacional, a mensagem JMS será armazenada no Gerenciador de Evento com Falha por padrão para exclusão ou reprodução manual.

Nota: É possível alterar o modo de recuperação na ligação de modo que a mensagem seja retrocedida ao invés de ser armazenada no gerenciador de evento com falha.

- Se o fluxo de mediação não for transacional, a exceção será registrada e a mensagem será perdida.

A situação é semelhante para um manipulador de dados. Como o manipulador de dados é chamado pela ligação de dados, uma exceção do manipulador de dados é produzida em uma exceção de ligação de dados. Portanto, uma **DataHandlerException** é reportada para você como uma **DataBindingException**.

Limitações de Ligações

As ligações possuem algumas limitações em seus usos que estão listadas aqui.

Limitações da Ligação do MQ:

As ligações do MQ têm algumas limitações em seu uso que são listadas aqui.

Nenhuma Distribuição de Mensagem de Publicação-Assinatura

O método de distribuição de mensagens de publicação-assinatura atualmente não é suportado pela ligação do MQ, embora o WMQ em si suporte publicação-assinatura. No entanto, a ligação JMS do MQ suporta este método de distribuição.

Filas de Recebimento Compartilhadas

Diversas ligações de importação e exportação do WebSphere MQ esperam que quaisquer mensagens presentes em suas filas de recebimento configuradas sejam destinadas para essa exportação ou importação. Ligações de importação e exportação devem ser configuradas com as considerações a seguir:

- Cada importação do MQ deve ter uma fila de recebimento diferente porque a ligação de importação do MQ assume que todas as mensagens na fila de recebimento são respostas às solicitações que ela envia. Se a fila de recebimento for compartilhada por mais de uma importação, as respostas poderão ser recebidas pela importação incorreta e não serão correlacionadas com a mensagem de solicitação original.
- Cada exportação do MQ deve ter uma fila de recebimento diferente, porque, do contrário, você não pode prever qual exportação obterá qualquer mensagem de solicitação específica.
- Importações e exportações do MQ podem apontar para a mesma fila de envio.

Limitações das Ligações JMS, JMS do MQ e JMS Genéricas:

As ligações JMS e JMS do MQ possuem algumas limitações.

Implicações da Geração de Ligações Padrão

As limitações de uso das ligações JMS, JMS do MQ e JMS genéricas são discutidas nas seguintes seções:

- Implicações da Geração de Ligações Padrão
- Esquema de Correlação de Resposta
- Suporte Bidirecional

Quando você gerar uma ligação, vários campos serão preenchidos para você como os padrões, se não escolher inserir os valores você mesmo. Por exemplo, um nome de connection factory será criado para você. Se você souber que colocará seu aplicativo em um servidor e o acessará remotamente com um cliente, deverá, no momento da criação da ligação, inserir os nomes JNDI em vez de usar os padrões, pois você provavelmente deseja controlar esses valores através do console administrativo no tempo de execução.

No entanto, se você aceitou os padrões e descobriu posteriormente que não pode acessar seu aplicativo a partir de um cliente remoto, será possível usar o console administrativo para configurar explicitamente o valor da connection factory. Localize o campo de terminais do provedor nas configurações de connection factory e inclua um valor tal como <server_hostname>:7276 (se estiver usando o número da porta padrão).

Esquema de Correlação de Resposta

Se você usar o esquema de correlação de resposta CorrelationId To CorrelationId, usado para correlacionar mensagens em uma operação de solicitação-resposta, deverá ter um ID de correlação dinâmico na mensagem.

Para criar um ID de correlação dinâmico em um módulo de mediação usando o editor de fluxo de mediação, inclua uma primitiva de mediação Mapeamento antes da importação com a ligação JMS. Abra o editor de mapeamento. Os cabeçalhos de arquitetura de componente de serviço conhecidos estarão disponíveis na mensagem de destino. Arraste um campo que contém um ID exclusivo na mensagem de origem até o ID de correlação no cabeçalho JMS na mensagem de destino.

Suporte Bidirecional

Apenas caracteres ASCII são suportados para Java Naming and Directory Interface (JNDI) nomes no tempo de execução.

Filas de Recebimento Compartilhadas

Diversas Ligações de exportação e importação esperam que quaisquer mensagens presentes em suas filas de recebimento configuradas sejam destinadas para essa exportação ou importação. Ligações de importação e exportação devem ser configuradas com as considerações a seguir:

- Cada ligação de importação deve ter uma fila de recebimento diferente porque a ligação de importação assume que todas as mensagens na fila de recebimento são respostas para solicitações que são enviadas. Se a fila de recebimento for compartilhada por mais de uma importação, as respostas poderão ser recebidas pela importação incorreta e não serão correlacionadas com a mensagem de solicitação original.
- Cada exportação deve ter uma fila de recebimento diferente pois, caso contrário, você não poderá prever qual exportação obterá qualquer mensagem de solicitação específica.
- Importações e exportações podem apontar para a mesma fila de envio.

Objetos de Negócios

A indústria do software de computador desenvolveu vários modelos e estruturas de programação em que *objetos de negócios* fornecem uma representação natural dos dados de negócios para processamento do aplicativo.

Em geral, estes objetos de negócios:

- São definidos usando padrões de mercado
- Mapeiam dados de forma transparente para tabelas de banco de dados ou Enterprise Information Systems
- Suportam protocolos de chamada remota
- Fornecem a base do modelo de programação para programação de aplicativo

O Process Designer e o Integration Designer fornecem aos desenvolvedores um tipo de modelo de objeto de negócios comum para representar os diferentes tipos de entidades de negócios de diferentes domínios.

No Process Designer, o foco dos objetos de negócios é uma representação de tipo de dados. Os objetos de negócios de base (tipos de variáveis) são fornecidos nos kits de ferramentas do sistema e você pode criar tipos de variáveis customizados chamados objetos de negócios customizados. Consulte *Objetos de Negócios e Variáveis*.

No Integration Designer, que está disponível apenas com o IBM Business Process Manager Advanced, os objetos de negócios podem representar construções XSD mais complexas. No Integration Designer, os objetos de negócios possuem uma afinidade próxima com esquemas XML. Para obter informações sobre o que deve ser considerado ao integrar objetos de negócios definidos no Integration Designer a objetos de negócios definidos no Process Designer, consulte *Espelhamento da Biblioteca e Construções XML não Suportadas*.

No momento do desenvolvimento no Integration Designer, o modelo de objeto de negócios permite que os desenvolvedores definam objetos de negócios como definições de esquema XML. No tempo de execução, os dados de negócios definidos pelas definições do esquema XML são representados como *Objetos de Negócios Java*. Neste modelo, objetos de negócios são baseados livremente em rascunhos antigos da especificação de Objeto de Dados de Serviço (SDO) e fornecem o conjunto completo de interfaces de aplicativo do modelo de programação necessárias para manipular dados de negócios. Os subtópicos a seguir descrevem melhor como é possível usar objetos de negócios com o Integration Designer.

Definindo objetos de negócios

Você pode definir objetos de negócios usando o editor de objeto de negócios no Integration Designer. O editor de objeto de negócios armazena os objetos de negócios como definições de esquema XML.

O uso do esquema XML para definir objetos de negócios fornece várias vantagens:

- O esquema XML fornece um modelo de definição de dados baseado em padrões e uma base para interoperabilidade entre diferentes sistemas e aplicativos heterogêneos. O esquema XML é usado em conjunto com o Web Services Description Language (WSDL) para fornecer contratos de interface baseados em padrões entre componentes, aplicativos e sistemas.
- O esquema XML define um modelo de definição de dados rico para representar dados de negócios. Este modelo inclui tipos complexos, tipos simples, tipos definidos pelo usuário, herança de tipo e cardinalidade, entre outros recursos.
- Objetos de negócios podem ser definidos por interfaces de negócios e dados definidos no Web Services Description Language, bem como por esquema XML a partir de organizações padrões de mercado ou de outros sistemas e aplicativos. O Integration Designer pode importar esses objetos de negócios diretamente.

O Integration Designer também fornece suporte para descobrir os dados de negócios em bancos de dados e Enterprise Information Systems e, em seguida, gerar a definição de objeto de negócios de esquema XML baseado em padrões desses dados de negócios. Objetos de negócios gerados desta maneira são geralmente referidos como *objetos de negócios específicos de aplicativo* porque eles imitam a estrutura dos dados de negócios definidos no Enterprise Information System.

Quando um processo está manipulando dados de muitos sistemas de informações diferentes, pode ser útil transformar a representação diferente de dados de negócios (por exemplo, CustomerEIS1 e CustomerEIS2 ou OrderEIS1 e OrderEIS2) em uma representação canônica única (por exemplo, Cliente ou Pedido). A representação canônica é muitas vezes referida como o *objeto de negócios genérico*.

Definições de objeto de negócios, especialmente para objetos de negócios genéricos, são usadas com frequência por mais de um aplicativo. Para suportar essa reutilização, o Integration Designer permite que os objetos de negócios sejam criados nas bibliotecas que podem, então, ser associadas a diversos módulos de aplicativos.

O Web Services Description Language (WSDL) define os contratos para os serviços fornecidos e consumidos por um módulo aplicativo Service Component Architecture (SCA), bem como os contratos usados para criar os componentes dentro de um módulo aplicativo. Em um contrato, um WSDL pode representar as operações e os objetos de negócios (que são definidos pelo esquema XML para representar os dados de negócios).

Trabalhando com objetos de negócios

O Service Component Architecture (SCA) fornece a estrutura para definir um módulo aplicativo, os serviços que ele fornece, os serviços que ele consome e a composição de componentes que fornecem a lógica de negócios do módulo aplicativo. Objetos de negócios desempenham um papel importante no aplicativo, definindo os dados de negócios que são usados para descrever o serviço e contratos de componente e os dados de negócios que os componentes manipulam.

O diagrama a seguir descreve um módulo aplicativo SCA e ilustra muitos dos locais em que o desenvolvedor trabalha com objetos de negócios.

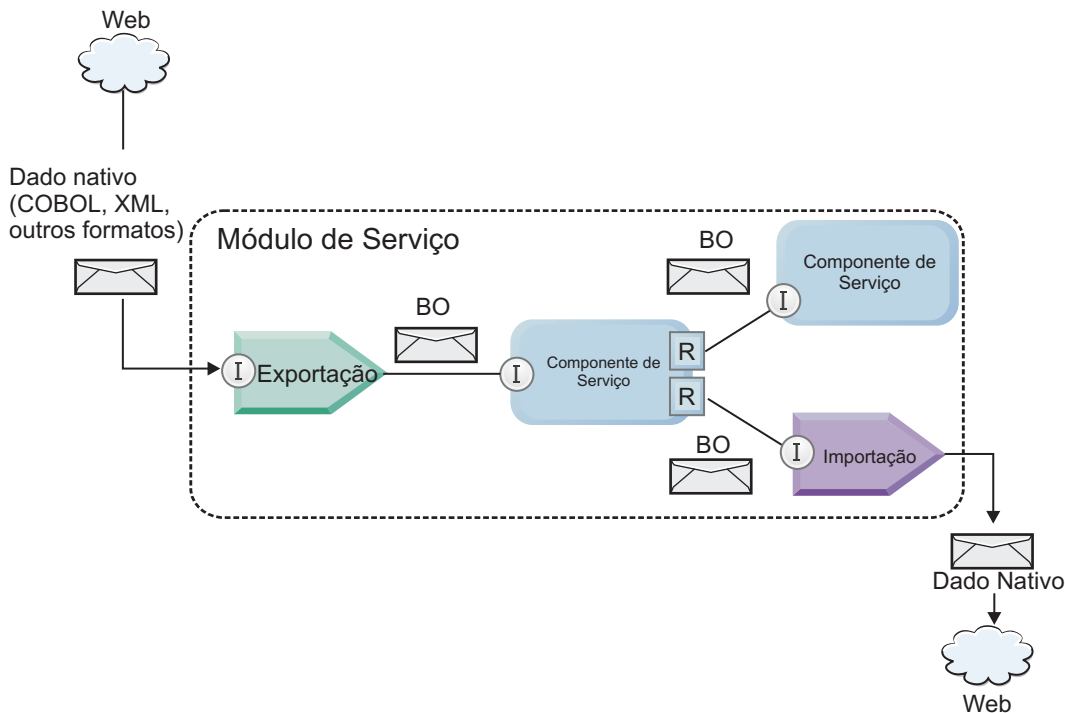


Figura 86. Objetos de Negócios Representam os Dados que Fluem entre Serviços em um Aplicativo

Nota: Este tópico descreve como os objetos de negócios são usados pelos módulos de aplicativo SCA. Se estiver usando interfaces Java, os módulos de aplicativo SCA também poderão processar objetos Java.

Modelo de Programação de Objeto de Negócios

O modelo de programação do objeto de negócios consiste em um conjunto de interfaces Java que representam:

- A definição do objeto de negócios e dados de instância
- Um conjunto de serviços que suportam as operações nos objetos de negócios

As definições de tipo de objeto de negócios são representadas pelas interfaces `commonj.sdo.Type` e `commonj.sdo.Property`. O modelo de programação de objeto de negócios fornece um conjunto de regras para mapear as informações de tipo complexo de esquema XML para a interface de Tipo e cada um dos elementos na definição de tipo complexo para a interface de Propriedade.

Instâncias de objetos de negócios são representadas pela interface `commonj.sdo.DataObject`. O modelo de programação do objeto de negócios é sem tipo, o que significa que a mesma interface `commonj.sdo.DataObject` pode ser usada para representar as definições de objeto de negócios diferentes, tais como Cliente e Pedido. A definição de quais propriedades podem ser configuradas e recuperadas a partir de cada objeto de negócios é determinada pelas informações de tipo definidas no esquema XML associado a cada objeto de negócios.

O comportamento do modelo de programação do objeto de negócios é baseado na especificação Service Data Object 2.1. Para obter informações adicionais, consulte a especificação SDO 2.1 para Java, tutoriais e javadocs na web: <http://www.oasis-open.org/>.

Serviços de objeto de negócios suportam várias operações de ciclo de vida (como criação, igualdade, análise e serialização) nos objetos de negócios.

Para informações específicas sobre o modelo de programação do objeto de negócio, consulte Programando usando serviços de objeto de negócio e Documentação API e SPI gerada nos objetos de negócio.

Ligações, Ligações de Dados e Manipuladores de Dados

Conforme mostrado em Figura 41 na página 155, os dados de negócios que são usados para chamar serviços fornecidos pelos módulos de aplicativo SCA são transformados em objetos de negócios para que os componentes SCA possam manipular os dados de negócios. De modo semelhante, os objetos de negócios manipulados pelos componentes SCA são convertidos no formato de dados requerido pelos serviços externos.

Em alguns casos, tal como a ligação de serviço da Web, a ligação usada para exportar e importar serviços transforma automaticamente os dados no formato apropriado. Em outros casos, tal como a ligação JMS, os desenvolvedores podem fornecer uma ligação de dados ou o manipulador de dados que converte formatos não nativos em objetos de negócios representados pela interface DataObject.

Para obter informações adicionais sobre o desenvolvimento de ligações de dados e manipuladores de dados, consulte “Manipuladores de Dados” na página 59 e “Ligações de Dados” na página 61.

Componentes

Os componentes SCA definem seus contratos de serviço de provisão e consumo usando uma combinação do esquema Web Services Description Language e XML. Os dados de negócios que o SCA transmite entre componentes são representados como objetos de negócios usando a interface DataObject. O SCA verifica se esses tipos de objeto de negócios são compatíveis com o contrato de interface definido pelo componente a ser chamado.

As abstrações de modelo de programação para manipulação de objetos de negócios variam de componente para componente. O componente POJO e a primitiva Customizada do componente de fluxo de mediação fornecem a manipulação direta dos objetos de negócios ativando a programação Java usando diretamente as interfaces de programação e os serviços de objeto de negócios. A maioria dos componentes fornece abstrações de nível superior para a manipulação de objetos de negócios, mas também fornecem fragmentos de código Java para definir o comportamento customizado nas interfaces e serviços de objeto de negócios.

Os objetos de negócios podem ser transformados usando a combinação da Mediação do Fluxo de Interface e do componente Mapa de Objeto de Negócios ou do componente do fluxo de mediação e sua primitiva de Mapa XML. Esses recursos de transformação do objeto de negócios são úteis para converter objetos de negócios específicos do aplicativo para e a partir de objetos de negócios genéricos.

Objetos de Negócios Especiais

Os objetos de mensagem de serviço e gráficos de negócios são dois tipos especializados de objetos de negócios que são usados para propósitos de aplicativo específicos.

Objeto de mensagem de serviço

Um objeto de mensagem de serviço (SMO) é um objeto de negócios especializado que é usado pelos componentes de fluxo de mediação para representar a coleção de dados associados a uma chamada de serviço.

Um SMO possui uma estrutura fixa de nível superior que consiste em cabeçalhos, contexto, corpo e anexos (se presente).

- Os cabeçalhos transportam informações relacionadas à chamada de serviço sobre um protocolo ou uma ligação específica. Os exemplos são cabeçalhos SOAP e cabeçalhos JMS.

- Dados de contexto transportam informações lógicas adicionais associadas à chamada enquanto ela está sendo processada pelo componente do fluxo de mediação. Estas informações geralmente não fazem parte dos dados do aplicativo enviados ou recebidos por clientes.
- O corpo do SMO transporta os dados de negócios de carga útil, que representam os principais dados da mensagem do aplicativo ou da chamada na forma de um objeto de negócios padrão.

O SMO também pode transportar dados de anexo para chamadas de serviço da web usando SOAP com anexos.

Fluxos de mediação executam tais tarefas como roteamento de solicitação e transformação de dados e o SMO fornece a visualização combinada de conteúdo de cabeçalho e de carga útil em uma estrutura unificada única.

Gráfico de Negócios

Um gráfico de negócios é um objeto de negócios especial usado para fornecer suporte para sincronização de dados em cenários de integração.

Considere um exemplo no qual dois sistemas de informações corporativos possuem uma representação de um pedido específico. Quando o pedido é alterado em um sistema, uma mensagem pode ser enviada para o outro sistema para sincronizar os dados do pedido. Os gráficos de negócios suportam a noção de envio apenas na parte do pedido que foi alterada no outro sistema e anotando-a com informações de resumo da mudança para definir o tipo de alteração.

Neste exemplo, um gráfico de negócios Pedido seria transmitido para o outro sistema em que um dos itens de linha no pedido foi excluído e que a propriedade de data de envio projetada do pedido foi atualizada.

Os gráficos de negócios podem ser facilmente incluídos em objetos de negócios existentes no Integration Designer. Eles estão mais frequentemente localizados em cenários nos quais adaptadores do WebSphere estão sendo usados e para suportar a migração dos aplicativos do WebSphere InterChange Server.

Modo de Análise de Objeto de Negócios

O Integration Designer fornece uma propriedade em módulos e bibliotecas que você pode usar para configurar o modo de análise de XML para objetos de negócios para mais rápido ou lento.

- Se a opção for configurada como *mais rápido*, fluxos de bytes são XML serão analisados mais rápido para criar o objeto de negócios.
- Se a opção for configurada como *lento*, o objeto de negócios será criado normalmente, mas a análise real do fluxo de byte XML será adiada e parcialmente analisada apenas quando as propriedades do objeto de negócios forem acessadas.

No modo de análise XML, dados não XML são sempre analisados rapidamente para criar o objeto de negócios.

Considerações ao Escolher o Modo de Análise do Objeto de Negócios

O modo de análise de objeto de negócios determina como os dados XML são analisados no tempo de execução. Um modo de análise de objeto de negócios é definido em um módulo ou uma biblioteca quando ela é criada. É possível alterar o modo de análise do módulo ou da biblioteca, no entanto, você deve estar ciente das implicações.

O modo de análise de objeto de negócios é configurado no nível do módulo e da biblioteca. Os módulos que foram criados em uma versão do IBM Integration Designer anterior à versão 7 serão executados no modo de análise mais rápido sem qualquer alteração necessária. Por padrão, os módulos e bibliotecas que são criados no IBM Integration Designer versão 7 e em versões mais recentes receberão o modo de análise mais adequado dependendo de vários fatores, tal como o modo de análise de projetos existentes

em sua área de trabalho ou o modo de análise de projetos dependentes ou outros projetos na mesma solução e assim por diante. É possível alterar o modo de análise do objeto de negócios de um módulo ou biblioteca para adequar sua implementação, no entanto, você deve estar ciente das considerações a seguir.

Considerações

- O modo de análise de objeto de negócios lento processa dados XML mais rápido; entretanto, há diferenças de compatibilidade entre o modo mais rápido e o modo lento que você precisa saber antes de alterar a configuração de um módulo ou uma biblioteca. Essas diferenças irão afetar o comportamento de tempo de execução dos módulos. Para obter informações sobre qual modo de análise é ideal para seu aplicativo, consulte "Benefícios do uso do modo de análise lenta versus rápida" nos links relacionados.
- Um módulo pode ser configurado apenas para executar em um modo de análise. As bibliotecas podem ser configuradas para suportar qualquer um dos modos de análise ou ambos os modos de análise. Uma biblioteca que é configurada para suportar ambos os modos de análise pode ser referenciada por um módulo usando o modo de análise mais rápido e um módulo usando o modo de análise lento. O modo de análise de uma biblioteca no tempo de execução é determinado pelos módulos que fazem referência à biblioteca. No tempo de execução, um módulo declara seu modo de análise e esse modo de análise é usado pelo módulo e quaisquer bibliotecas que o módulo usa.
- Módulos e bibliotecas que estão configurados para modos de análise diferentes são compatíveis nos seguintes casos:
 - Módulos e bibliotecas configurados com o modo de análise lento são compatíveis com bibliotecas que usam o modo de análise lento ou ambos os modos de análise, mais rápido e lento.
 - Módulos e bibliotecas configurados com o modo de análise mais rápido são compatíveis com bibliotecas que usam o modo de análise mais rápido ou ambos os modos de análise, mais rápido e lento.
 - Bibliotecas configuradas com os modos de análise lento e mais rápido são compatíveis apenas com bibliotecas que usam ambos os modos de análise, lento e mais rápido.
- Use o mesmo modo de análise para interagir os módulos que se comunicam usando a ligação SCA. Se os módulos se comunicarem usando modos de análise diferentes, isto poderá resultar em problemas de desempenho.

Conceitos relacionados:

"Benefícios de Usar o Modo de Análise Lento Versus Mais Rápido" na página 158

Alguns aplicativos se beneficiam do modo de análise de XML lento, enquanto outros veem o desempenho melhorado com modo de análise mais rápido. É recomendado que você teste o desempenho de seu aplicativo em ambos os modos de análise para determinar quais modo melhor se ajusta às características específicas de seu aplicativo.

Benefícios de Usar o Modo de Análise Lento Versus Mais Rápido

Alguns aplicativos se beneficiam do modo de análise de XML lento, enquanto outros veem o desempenho melhorado com modo de análise mais rápido. É recomendado que você teste o desempenho de seu aplicativo em ambos os modos de análise para determinar quais modo melhor se ajusta às características específicas de seu aplicativo.

Aplicativos que analisam grandes fluxos de dados XML provavelmente verão as melhorias de desempenho quando o modo de análise XML lento é usado. Os benefícios de desempenho aumentam à medida que o tamanho do fluxo de byte XML aumenta e a quantidade de dados do fluxo de byte que é acessado pelo aplicativo diminui.

Os aplicativos a seguir provavelmente são executados melhor usando um modo de análise rápido:

- Aplicativos que analisam os fluxos de dados não XML
- Aplicativos que usam mensagens que são criadas usando o serviço BOFactory
- Aplicativos que analisam mensagens XML muito pequenas

Referências relacionadas:

“Considerações ao Escolher o Modo de Análise do Objeto de Negócios” na página 157

O modo de análise de objeto de negócios determina como os dados XML são analisados no tempo de execução. Um modo de análise de objeto de negócios é definido em um módulo ou uma biblioteca quando ela é criada. É possível alterar o modo de análise do módulo ou da biblioteca, no entanto, você deve estar ciente das implicações.

Considerações sobre Migração e Desenvolvimento de Aplicativo

Se estiver configurando um aplicativo que foi desenvolvido originalmente usando o modo de análise mais rápido para usar agora o modo de análise lento, ou se estiver planejando para alternar um aplicativo entre o modo de análise lento ou mais rápido, esteja ciente das diferenças entre modos e as considerações ao alternar os modos.

Identificador de Erros

Se o fluxo de byte XML que está sendo analisado estiver malformatado, ocorrerão exceções de análise.

- No modo de análise de XML mais rápido, essas exceções ocorrem assim que o objeto de negócios é analisado a partir do fluxo de XML de entrada.
- Se o modo de análise de XML lento estiver configurado, as exceções de análise ocorrerão latentemente quando as propriedades do objeto de negócios forem acessadas e a parte do XML que está malformatada for analisada.

Para lidar com XML malformatado, selecione uma das opções a seguir:

- Implemente um barramento de serviço corporativo nas extremidades para validar o XML de entrada
- Crie a lógica de detecção de erro lenta no ponto em que as propriedades de objeto de negócios são acessadas

Pilhas e Mensagens de Exceções

Como os modos de análise de XML mais rápido e lento têm diferentes implementações subjacentes, os rastreios de pilha lançados pelas interfaces de programação e serviços de objeto de negócios possuem o mesmo nome da classe de exceção, mas eles podem não conter a mesma mensagem de exceção ou conjunto agrupado de classes de exceção específicas da implementação.

Formato de Serialização XML

O modo de análise de XML lento fornece uma otimização de desempenho que tenta copiar XML não modificado a partir do fluxo de bytes de entrada para o fluxo de bytes de saída na serialização. O resultado é o desempenho aprimorado, mas o formato de serialização do fluxo de byte XML de saída pode ser diferente se o objeto de negócios inteiro foi atualizado no modo de análise de XML lento ou se ele estava em execução no modo de análise de XML mais rápido.

Embora o formato de serialização de XML possa não ser precisamente sintaticamente equivalente, o valor semântico fornecido pelo objeto de negócios é equivalente independente dos modos de análise e XML pode ser seguramente transmitido entre os aplicativos em execução em diferentes modos de análise com equivalência de semântica.

Validador de Instância do Objeto de Negócios

O validador da instância do modo de objeto de negócios de análise de XML fornece uma validação com maior fidelidade de objetos de negócios, particularmente no aspecto da validação de valores da propriedade. Devido a essas melhorias, o validador de instância de modo de análise lento captura problemas adicionais que não são capturados no modo de análise mais rápido e fornece mensagens de erro mais detalhadas.

Mapas XML Versão 602

Os fluxos de mediação originalmente desenvolvidos antes do WebSphere Integration Developer Versão 6.1 podem conter primitivas de Mapeamento que usam um mapa ou folha de estilo que não pode ser executados diretamente em modo de análise XML lento. Quando um aplicativo é migrado para uso em modo de análise de XML lento, os arquivos de mapa associados a primitivas de mapeamento podem ser automaticamente atualizados pelo assistente de migração para executar no novo modo. Entretanto, se uma primitiva Mapeamento referir-se diretamente a uma folha de estilo que foi editada manualmente, a folha de estilo não será migrada e não poderá ser executada em modo de análise XML lento.

APIs Não Publicadas Privadas

Se um aplicativo estiver tirando vantagem de interfaces de programação de objeto de negócios específicas da implementação, não publicadas, privadas, o aplicativo provavelmente falhará na compilação quando o modo de análise for alternado. No modo de análise mais rápido, essas interfaces privadas serão geralmente as classes de implementação de objeto de negócios definidas pelo Eclipse Modeling Framework (EMF).

Em todos os casos, é altamente recomendado que APIs privadas sejam removidas do aplicativo.

APIs do EMF do Objeto de Mensagem de Serviço

Um componente de mediação no IBM Integration Designer fornece a capacidade de manipular o conteúdo da mensagem usando as classes e interfaces Java fornecidas no pacote `com.ibm.websphere.sibx.smobo`. No modo de análise de XML lento, as interfaces Java no pacote `com.ibm.websphere.sibx.smobo` ainda podem ser usadas, mas os métodos que se referem diretamente às classes e interfaces do Eclipse Modeling Framework (EMF) ou que são herdados de interfaces do EMF provavelmente falharão.

O `ServiceMessageObject` e seu conteúdo não podem ser convertidos em objetos EMF no modo de análise de XML lento.

Serviço BOMode

O serviço BOMode é usado para determinar se o modo de análise de XML atualmente em execução é mais rápido ou lento.

Migração

Todos os aplicativos anteriores à versão 7.0.0.0 estão em execução no modo de análise mais rápido. Quando eles são migrados no tempo de execução usando as ferramentas de migração de tempo de execução BPM, eles continuam executando no modo de análise de XML mais rápido.

Para permitir que um aplicativo anterior à versão 7.0.0.0 seja configurado para usar o modo de análise de XML lento, primeiro você usa o Integration Designer para migrar os artefatos do aplicativo. Após a migração, você configura o aplicativo para usar a análise de XML lenta.

Consulte *Migrando Artefatos de Origem* para obter informações sobre como migrar artefatos no Integration Designer e consulte *Configurando o Modo de Análise de Objeto de Negócios de Módulos e Bibliotecas* para obter informações sobre como configurar o modo de análise.

Relacionamentos

Um relacionamento é uma associação entre duas ou mais entidades de dados, geralmente objetos de negócios. No IBM Business Process Manager Advanced, os relacionamentos podem ser usados para transformar dados equivalentes em objetos de negócios e outros dados, mas isso é representado de forma diferente ou pode ser usados para traçar associações por meio de diferentes objetos localizados em diferentes aplicativos. Eles podem ser compartilhados entre os aplicativos, soluções, e até mesmo em produtos.

O serviço de relacionamento no IBM Business Process Manager Advanced fornece a infra-estrutura e operações para gerenciar relacionamentos. Como ele permite que você lide com objetos de negócios independentemente de onde eles residem, ele pode fornecer uma visualização holística unificada em todos os aplicativos de uma empresa, e servem como um bloco de construção para soluções BPM. Como os relacionamentos são extensíveis e gerenciáveis, eles podem ser usados em soluções de integração complexas.

O Que São Relacionamentos?

Um relacionamento é uma associação entre objetos de negócios. Cada objeto de negócios em um relacionamento é chamado de *participante* no relacionamento. Cada participante no relacionamento é distinto dos outros participantes com base na função, ou *atribuição*, que ele serve nesse relacionamento. Um relacionamento contém uma lista de funções.

A *definição* de relacionamento descreve cada função e especifica como as funções são relacionadas. Também descreve o "formato" geral do relacionamento. Por exemplo, esta função pode ter apenas um participante, mas esta outra função pode tantos participantes quanto necessário. É possível definir um relacionamento *carro-proprietário*, por exemplo, onde um proprietário pode possuir diversos carros. Por exemplo, uma instância pode ter os seguintes participantes para cada uma destas funções:

- Carro (Ferrari)
- Proprietário (John)

A definição de relacionamento é um modelo para a *instância* de relacionamento. A instância é a instanciação do tempo de execução do relacionamento. No exemplo *car-owner*, uma instância pode descrever qualquer uma das seguintes associações:

- John possui Ferrari
- Sara possui Mazda
- Bob possui Ferrari

O uso de relacionamentos o libera da necessidade de construir persistência de rastreamento de relacionamento customizada dentro de sua lógica de negócios. Para determinados cenários, o serviço de relacionamento faz todo o trabalho para você. Consulte o exemplo descrito na seção em Relacionamentos de Identidades.

Cenários

A seguir há um exemplo típico de uma situação na qual uma solução de integração pode usar relacionamentos. Uma grande corporação compra diversas empresas, ou unidades de negócios. Cada unidade de negócios usa um software diferente para monitorar a equipe e notebooks. A empresa precisa de uma forma de monitorar seus funcionários e notebooks. Ela quer uma solução que permita:

- Visualizar todos os funcionários nas várias unidades de negócios como se estivessem em um banco de dados
- Ter uma única visualização de todos os seus notebooks
- Permitir que os funcionários efetuem logon no sistema e comprem um notebook
- Acomodar os sistemas de aplicativos corporativos diferentes nas várias unidades de negócios

Para realizar isso, a empresa precisa de uma maneira de assegurar, por exemplo, que John Smith e John A. Smith em diferentes aplicativos sejam vistos como o mesmo funcionário. Por Exemplo, ela precisa de uma maneira de consolidar uma única entidade em diversos espaços do aplicativo.

Cenários de relacionamento mais complexos envolvem construir processos BPEL que desenham relacionamentos em diferentes objetos localizados em diversos aplicativos. Com cenários de relacionamento complexos, os objetos de negócios residem na solução de integração e não nos aplicativos. O serviço de relacionamento fornece uma plataforma para gerenciar relacionamentos de forma persistente. Antes do serviço de relacionamento, você precisaria construir seu próprio serviço de persistência de objeto. Dois exemplos de cenários de relacionamento complexos são:

- Você tem um objeto de negócios **carro** com um número VIN em um aplicativo SAP e deseja controlar o fato de que este carro é de propriedade de outra pessoa. No entanto, o relacionamento de propriedade está com alguém em um aplicativo PeopleSoft. Neste padrão de relacionamentos, você tem duas soluções e precisa construir uma ponte de ligação entre elas.
- Uma empresa de varejo grande deseja ser capaz de monitorar o merchandise retornado para dinheiro ou crédito. Há dois aplicativos diferentes envolvidos: um sistema de gerenciamento de pedidos (OMS) para compras e um sistema de gerenciamento devoluções (RMS) para devoluções. Os objetos de negócios residem em mais de um aplicativo e você precisa de uma maneira de mostrar os relacionamentos que existem entre eles.

Padrões de Uso Comuns

Os padrões de relacionamento mais comuns são padrões de *equivalência*. Eles são baseados em referência cruzada ou correlação. Há dois tipos de relacionamentos que se encaixam neste padrão: *sem identidade e identidade*.

- **Relacionamentos Sem Identidade** estabelecem associações entre objetos de negócios ou outros dados em uma base de um para muitos ou de muitos para muitos. Para cada instância de relacionamento, pode haver uma ou mais instâncias de cada participante. Um tipo de relacionamento de não identidade é um relacionamento de consulta estática. Um exemplo disso é um relacionamento no qual **CA** em um aplicativo SAP está relacionado com **California** em um aplicativo Siebel.

•

Relacionamentos de Identidades estabelecem associações entre objetos de negócios ou outros dados em uma base de um para um. Para cada instância de relacionamento, pode haver apenas uma instância de cada participante. Os relacionamentos de identidades capturam referências cruzadas entre objetos de negócios que são semanticamente equivalentes, mas que são identificados de maneira diferente em aplicativos diferentes. Cada participante do relacionamento está associado a um objeto de negócios que possui um valor (ou combinação de valores) que identifica exclusivamente o objeto. Relacionamentos de identidade geralmente transformam os atributos-chave de objetos de negócios, como números de ID e códigos do produto.

Por exemplo, se você tiver objetos de negócios **carro** nos aplicativos SAP, PeopleSoft e Siebel e desejar construir uma solução que os sincroniza, normalmente precisará introduzir a lógica de sincronização do relacionamento criada à mão em seis mapas:

SAP -> genérico

genérico -> SAP

PeopleSoft-> genérico

genérico-> PeopleSoft

Siebel-> Siebel

genérico-> Siebel

No entanto, se você usar relacionamentos em sua solução, o serviço de relacionamento fornecerá implementações padrão pré-construídas que mantêm todas essas instâncias de relacionamento para você.

Ferramentas para Trabalhar com Relacionamentos

O *editor de relacionamento* no Integration Designer é a ferramenta que você usa para modelar e projetar relacionamentos de integração de negócios e funções. Para obter informações complementares e de tarefas sobre a criação de relacionamentos e o uso do editor de relacionamentos, consulte Criando Relacionamentos.

O *serviço de relacionamento* é um serviço de infraestrutura no IBM Business Process Manager que mantém relacionamentos e funções no sistema e fornece operações para gerenciamento de relacionamentos e funções.

O *gerenciador de relacionamento* é a interface administrativa para gerenciar relacionamentos. Ele é acessado por meio das páginas do Gerenciador de Relacionamento do console administrativo.

Os relacionamentos podem ser chamados programaticamente por meio das APIs do serviço de relacionamento.

Serviço de Relacionamento

O serviço de relacionamento armazena dados do relacionamento em tabelas de relacionamentos, onde ele mantém o controle de valores específicos do aplicativo entre aplicativos e soluções. O serviço de relacionamento fornece operações para relacionamento e gerenciamento de função.

Como Funcionam os Relacionamentos

Relacionamentos e funções são definidos usando a interface gráfica da ferramenta editor de relacionamento no Integration Designer. O serviço de relacionamento armazena os dados de correlação em tabelas no banco de dados de relacionamento na origem de dados padrão que você especifica ao configurar o serviço de relacionamento. Uma tabela separada (às vezes chamada de tabela do participante) armazena informações para cada participante no relacionamento. O serviço de relacionamento usa essas tabelas de relacionamentos para manter o controle dos valores específicos do aplicativo relacionados e propagar informações atualizadas em todas as soluções.

Relacionamentos, que são artefatos de negócios, são implementados em um projeto ou em uma biblioteca compartilhada. Na primeira implementação, o serviço de relacionamento preenche os dados.

No tempo de execução, quando mapas ou outros componentes do IBM Business Process Manager precisam de uma instância de relacionamento, as instâncias do relacionamento são atualizadas ou recuperadas, dependendo do cenário.

Dados da instância de relacionamento e função podem ser manipulados por três meios:

- Chamadas de fragmento Java do componente IBM Business Process Manager das APIs de serviço de relacionamento
- Transformações de relacionamento no serviço de mapeamento de objeto de negócios do IBM Business Process Manager
- A ferramenta do gerenciador de relacionamentos

Para obter informações complementares e de tarefas detalhadas sobre a criação de relacionamentos, a identificação de tipos de relacionamento e o uso do editor de relacionamentos, consulte o tópico Criando Relacionamentos.

Gerenciador de Relacionamento

O gerenciador de relacionamentos é a interface administrativa para gerenciar relacionamentos. Ele é acessado por meio das páginas do Gerenciador de Relacionamento do console administrativo.

O gerenciador de relacionamentos fornece uma interface gráfica com o usuário para criação e manipulação de dados de relacionamento e função no tempo de execução. É possível gerenciar entidades de relacionamento em todos os níveis: níveis da instância de relacionamento, da instância de função e de dados do atributo e dados de propriedade. Com o gerenciador de relacionamentos, é possível:

- Visualizar uma lista dos relacionamentos no sistema e informações detalhadas para relacionamentos individuais
- Gerenciar instâncias de relacionamento:
 - Consultar dados de relacionamento para visualizar subconjuntos de dados de instância
 - Consultar dados de relacionamento para visualizar subconjuntos de dados de instância usando visualizações do banco de dados
 - Visualizar uma lista de instâncias de relacionamento que correspondem a uma consulta de relacionamento e informações detalhadas sobre uma instância
 - Editar os valores da propriedade para uma instância de relacionamento
 - Criar e excluir instâncias de relacionamento
- Gerenciar funções e instâncias de função:
 - Visualizar detalhes sobre uma função ou uma instância de função
 - Editar propriedades da instância de função
 - Criar e excluir instâncias de função para um relacionamento
 - Retroceder dados da instância de relacionamento para um ponto no tempo quando você sabe que os dados são confiáveis
- Importar dados de um relacionamento estático existente em seu sistema ou exportar dados a partir de um relacionamento estático existente para um arquivo CSV ou RI
- Remover esquema de relacionamento e dados do repositório quando o aplicativo que os usa é desinstalado

Relacionamentos em Ambientes de Implementação de Rede

Os relacionamentos podem ser usados nos ambientes de Implementação de Rede (ND) sem nenhuma configuração extra.

Nos ambientes de Implementação de Rede (ND), os relacionamentos são instalados em um cluster de aplicativo. Relacionamentos ficam, então, visíveis dentro do cluster e todos os servidores no cluster têm acesso aos dados de instância armazenados no banco de dados de relacionamento. A capacidade de executar o serviço de relacionamento em um ambiente de ND o torna escalável e altamente disponível.

O gerenciador de relacionamentos permite que relacionamentos sejam gerenciados em clusters diferentes por meio de uma interface administrativa centralizada. Você conecta o gerenciador de relacionamentos a um servidor em um cluster selecionando seu MBean de relacionamento.

APIs do Serviço de Relacionamento

Os relacionamentos podem ser chamados programaticamente por meio das APIs do serviço de relacionamento, dentro ou fora de mapas de objetos de negócios.

Três tipos de API estão disponíveis:

- APIs de manipulação de instância de relacionamento (incluindo criar, atualizar, excluir dados de instância diretamente)
- APIs de suporte padrão de relacionamento (incluindo `correlate()`, `correlateforeignKeyLookup`)
- Padrões de consulta de relacionamento (APIs de consulta)

O Barramento de Serviço Corporativo no IBM Business Process Manager

O IBM Business Process Manager suporta a integração de serviços de aplicativo, incluindo os mesmos recursos que o WebSphere Enterprise Service Bus.

Conectando Serviços Através de um Enterprise Service Bus

Com um barramento de serviço corporativo (ESB), é possível maximizar a flexibilidade de um SOA. Participantes em uma interação de serviço são conectados ao ESB, em vez de diretamente um com o outro.

Quando o solicitante de serviços se conecta ao ESB, o ESB assume a responsabilidade pela entrega de suas solicitações, usando mensagens, para um provedor de serviços que oferece a função e qualidade de serviço necessárias. O ESB facilita as interações entre solicitante-provedor e trata protocolos, padrões de interação ou recursos de serviço incompatíveis. Um ESB também pode ativar ou aprimorar o monitoramento e o gerenciamento. O ESB fornece recursos de virtualização e gerenciamento que implementam e estendem os recursos principais de SOA.

O ESB abstrai os seguintes recursos:

Local e Identidade

Os participantes não têm que conhecer o local ou identidade dos outros participantes. Por exemplo, os solicitantes não têm que estar ciente de que uma solicitação possa ser atendida por algum dos vários provedores; os provedores de serviço podem ser incluídos ou removidos sem interrupção.

Protocolo de Interação

Os participantes não têm que compartilhar o mesmo protocolo de comunicação ou estilo de interação. Por exemplo, uma solicitação expressa como SOAP sobre HTTP pode ser verificada por um provedor que só entende SOAP sobre Sistema de Mensagens Java (JMS).

Interface

Os solicitantes e provedores não têm que concordar com uma interface comum. Um ESB reconcilia as diferenças transformando as mensagens de solicitação e de resposta em um formato esperado pelo provedor.

Qualidades de Serviço (Interação)

Os participantes, ou administradores do sistema, declaram seus requisitos de qualidade de serviço, incluindo autorização de solicitações, criptografia e descriptografia de conteúdo da mensagem, auditoria automática de interações de serviço e como suas solicitações devem ser roteadas (por exemplo, otimizando para velocidade ou custo).

Interposição do ESB entre os participantes permite que você ajuste suas interações por meio de uma construção lógica chamada de *mediação*. As mediações operam em mensagens em andamento entre solicitantes e provedores. Por exemplo, as mediações podem ser usadas para localizar serviços com características específicas que um solicitante está solicitando ou para resolver diferenças de interface entre solicitantes e provedores. Para interações complexas, as mediações podem ser encadeadas sequencialmente.

Usando mediações, um barramento de serviço corporativo realiza as seguintes ações entre o solicitante e o serviço:

- *Roteamento* de mensagens entre os serviços. Um barramento de serviço corporativo oferece uma infraestrutura de comunicação comum que pode ser usada para conectar serviços e, portanto, as funções de negócios que eles representam, sem a necessidade de os programadores gravarem e manterem uma lógica de conectividade complexa.

- *Conversão* de protocolos de transporte entre o solicitante e o serviço. Um barramento de serviço corporativo fornece uma maneira consistente, baseada em padrões, para integrar funções de negócios que usam diferentes padrões de TI. Isso permite a integração de funções de negócios que normalmente não poderiam se comunicar, como conectar aplicativos em silos departamentais ou permitir que aplicativos em empresas diferentes participem em interações de serviços.
- *Transformação de* formatos da mensagem entre o solicitante e o serviço. O barramento de serviço corporativo permite que funções de negócios troquem informações em diferentes formatos, com o barramento assegurando que as informações entregues em uma função de negócios estejam no formato requerido por esse aplicativo.
- *Manipulação* de eventos de negócio a partir de origens diferentes. Um barramento de serviço corporativo suporta interações baseadas em eventos além das trocas de mensagens para manipular solicitações de serviço.

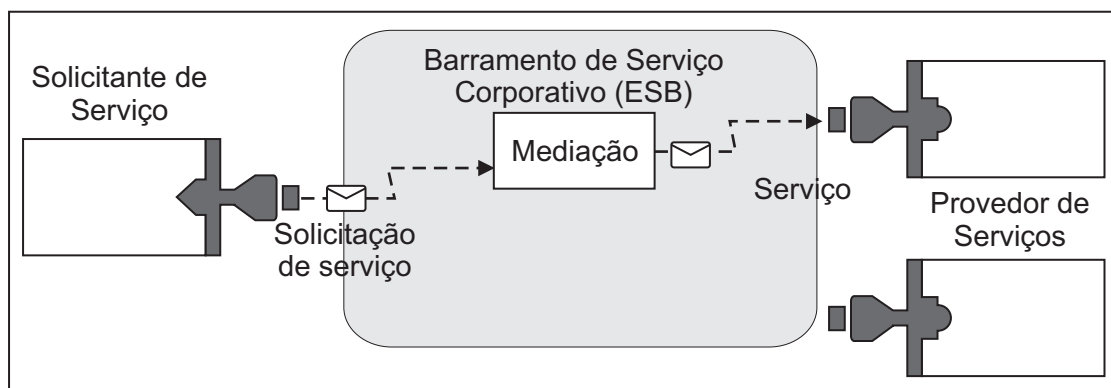


Figura 87. Um Barramento de Serviço Corporativo. O barramento de serviço corporativo está roteando as mensagens entre os aplicativos, os quais são solicitantes e provedores de serviços. O barramento está convertendo protocolos de transporte e transformando formatos da mensagem entre solicitantes e provedores. Nesta figura, cada aplicativo usa um protocolo diferente (representado pelos diferentes formatos geométricos de seus conectores) e usa diferentes formatos de mensagens.

Ao usar o barramento de serviço corporativo, é possível focar em seu negócio principal em vez de seus sistemas de computador. É possível alterar ou incluir nos serviços se necessário; por exemplo, para responder às mudanças da necessidade de negócios, para incluir capacidade extra de serviço ou incluir novos recursos. É possível fazer as alterações necessárias reconfigurando o barramento, com pouco ou nenhum impacto nos serviços e aplicativos existentes que usam o barramento.

Infraestrutura do Sistema de Mensagens do Barramento de Serviço Corporativo

O IBM Business Process Manager inclui recursos de barramento de serviço corporativo. O IBM Business Process Manager suporta a integração de tecnologias orientadas a serviço, orientadas a mensagem e conduzidas por evento para fornecer uma infraestrutura do sistema de mensagens baseada em padrões em um barramento de serviço corporativo integrado.

Os recursos de serviço corporativo que podem ser usados para seus aplicativos corporativos fornecem não apenas uma camada de transporte mas o suporte de mediação para facilitar as interações de serviço. O barramento de serviço corporativo é construído em torno de padrões abertos e arquitetura orientada a serviços (SOA). Ele é baseado na infraestrutura Java EE robusta e em serviços de plataforma associados fornecidos pelo IBM WebSphere Application Server Network Deployment.

O IBM Business Process Manager é desenvolvido com a mesma tecnologia disponível com o IBM WebSphere Enterprise Service Bus. Este recurso faz parte da funcionalidade subjacente do IBM Business Process Manager e nenhuma licença adicional para WebSphere Enterprise Service Bus é necessária para tirar vantagem destes recursos.

Entretanto, é possível implementar licenças independentes adicionais do WebSphere Enterprise Service Bus em sua empresa para estender o alcance da conectividade das soluções de integração de processos desenvolvidas com IBM Business Process Manager. Por exemplo, o WebSphere Enterprise Service Bus pode ser instalado mais próximo a um aplicativo SAP para hospedar um IBM WebSphere Adapter para SAP e transformar mensagens SAP antes de enviar essas informações na rede para um processo de negócios coreografado pelo IBM Business Process Manager.

Hosts do Sistema de Mensagens ou do Destino de Fila

Um host do sistema de mensagens ou do destino de fila fornece a função de sistema de mensagens dentro de um servidor. Um servidor torna-se o host de destino do sistema de mensagens quando você o configura como o destino do sistema de mensagens.

Um mecanismo do sistema de mensagens é executado em um servidor. O mecanismo do sistema de mensagens fornece funções do sistema de mensagens e um ponto de conexão para aplicativos se conectarem ao barramento. A comunicação assíncrona de SCA (Service Component Architecture), importações e exportações JMS e processamento interno assíncrono utilizam filas de mensagens no mecanismo do sistema de mensagens.

O ambiente de implementação conecta a origem da mensagem ao destino da mensagem através do barramento quando os módulos do aplicativo são implementados. Saber a origem da mensagem e o destino da mensagem ajuda você a determinar de que tipo de ambiente de implementação você precisa.

Aplicativos podem armazenar dados persistentes em um armazém de dados, que é um conjunto de tabelas em um banco de dados ou esquema, ou em um armazenamento de arquivos. O mecanismo do sistema de mensagens utiliza uma instância de uma origem de dados JDBC para interagir com esse banco de dados.

Configure o host de destino do sistema de mensagens ao definir seu ambiente de implementação utilizando o **Servidor** do console administrativo ou designe o servidor como o host de destino durante a instalação do software.

Armazém de Dados:

Cada mecanismo do sistema de mensagens pode utilizar um armazém de dados, que é um conjunto de tabelas em um banco de dados ou esquema que armazena dados persistentes.

Todas as tabelas no armazém de dados são mantidas no mesmo esquema do banco de dados. Você pode criar cada armazém de dados em um banco de dados separado. Como alternativa, pode criar vários armazéns de dados no mesmo banco de dados, com cada armazém de dados utilizando um esquema diferente.

Um mecanismo do sistema de mensagens utiliza uma instância de uma origem de dados JDBC para interagir com o banco de dados que contém o armazém de dados para esse mecanismo do sistema de mensagens.

Provedores JDBC

Você pode usar provedores JDBC para que haja interação dos aplicativos com os bancos de dados relacionais.

Os aplicativos utilizam provedores JDBC para interagir com os bancos de dados relacionais. O provedor JDBC fornece a classe de implementação específica do driver JDBC para acesso a um tipo de banco de dados específico. Para criar um conjunto de conexões para esse banco de dados, você associa uma origem de dados ao provedor JDBC. Em conjunto, os objetos do provedor JDBC e da origem de dados são funcionalmente equivalentes ao connection factory de Java EE Connector Architecture (JCA), que fornece conectividade com um banco de dados não relacional.

Consulte os exemplos da configuração de ambiente independente Típica e da configuração de ambiente de implementação Típica no tópico anterior.

Para obter mais informações sobre provedores JDBC, consulte “Provedores JDBC” no centro de informações do WebSphere Application Server.

Barramento de Integração de Serviço para IBM Business Process Manager

Um barramento de integração de serviços é um mecanismo de comunicação gerenciada que suporta a integração de serviços por meio de sistemas de mensagens síncronos e assíncronos. Um barramento consiste na interconexão de mecanismos de sistemas de mensagens que gerenciam recursos de barramento. É uma das tecnologias do WebSphere Application Server na qual o IBM Business Process Manager é baseado.

Um único barramento de integração de serviço e um único mecanismo do sistema de mensagens usam o mesmo esquema de banco de dados como o banco de dados do produto por padrão. Cada ambiente de implementação tem seu próprio barramento. Esse barramento único é chamado **BPM.deployment_environment_name.Bus**.

Um destino de barramento é um endereço lógico ao qual os aplicativos podem conectar-se como um produtor, consumidor ou ambos. Um destino de fila é um destino de barramento utilizado para o sistema de mensagens ponto a ponto.

Cada barramento pode ter um ou mais membros de barramento, cada um sendo um servidor ou um cluster.

A *topologia de barramento* é a organização física dos servidores de aplicativos, mecanismos do sistema de mensagens e gerenciadores de fila do WebSphere MQ e o padrão de conexões de barramento e links entre elas, que constitui seu barramento de serviço corporativo.

Aplicativos de Serviço e Módulos de Serviço

Um módulo de serviço é um módulo de Service Component Architecture (SCA) que fornece serviços no ambiente de tempo de execução. Ao implementar um módulo de serviço no IBM Business Process Manager, construa um aplicativo de serviço associado empacotado como um arquivo enterprise archive (EAR).

Os módulos de serviço são as unidades básicas de implementação e podem conter componentes, bibliotecas e módulos de temporariedade usados pelo aplicativo de serviço associado. Os módulos de serviço possuem exportações e, opcionalmente, importações para definir os relacionamentos entre módulos e solicitantes e provedores de serviço. O WebSphere Process Server suporta módulos para serviços de negócios e módulos de mediação. Ambos os módulos e módulos de mediação são tipos de módulos SCA. Um módulo de mediação permite a comunicação entre aplicativos transformando a chamada de serviço em um formato entendido pelo destino, transmitindo a solicitação ao destino e retornando o resultado ao originador. Um módulo para um serviço de negócios implementa a lógica de um processo de negócios. Entretanto, um módulo também pode incluir a mesma lógica de mediação que pode ser empacotada em um módulo de mediação.

Implementando um Aplicativo de Serviço

O processo de implementação de um arquivo EAR que contém um aplicativo de serviços é o mesmo que o processo de implementação de qualquer arquivo EAR. É possível modificar os valores para os parâmetros de mediação no momento da implementação. Após ter implementado um arquivo EAR contendo um módulo SCA, você pode visualizar detalhes sobre o aplicativo de serviço e seu módulo associado. É possível ver como um módulo de serviço é conectado aos solicitantes de serviços (por meio de exportações) e provedores de serviços (por meio de importações).

Visualizando Detalhes do Módulo SCA

Os detalhes do módulo de serviço que você pode visualizar dependem do módulo SCA. Eles incluem os seguintes atributos.

- Nome do módulo SCA
- Descrição do módulo SCA
- Nome do aplicativo associado
- Informações sobre a versão do módulo SCA, se o módulo estiver com versão
- Importações do módulo SCA:
 - Interfaces de importação são definições abstratas que descrevem como um módulo SCA acessa um serviço.
 - Ligações de importação são definições concretas que especificam o mecanismo físico pelo qual um módulo SCA acessa um serviço. Por exemplo, usando SOAP/HTTP.
- Exportações de módulo SCA:
 - Interfaces de exportação são definições abstratas que descrevem como solicitantes de serviços acessam um módulo SCA.
 - Ligações de exportação são definições concretas que especificam o mecanismo físico pelo qual um solicitante de serviços acessa um módulo SCA e, indiretamente, um serviço.
- Propriedades do Módulo SCA

Importações e Ligações de Importações

As importações definem as interações entre os módulos SCA e os fornecedores de serviços. Os módulos SCA usam importações para permitir que componentes acessem serviços externos (serviços que estão fora do módulo SCA) usando uma representação local. Ligações de importação definem a forma específica que um serviço externo é acessado.

Se os módulos SCA não precisarem acessar serviços externos, eles não precisarão ter importações. Os módulos de mediação normalmente uma ou mais importações que são utilizadas para passar mensagens ou solicitações a seus destinos pretendidos.

Interfaces e Ligações

Uma importação do módulo SCA precisa de pelo menos uma interface e uma importação do módulo SCA possui uma única ligação.

- Interfaces de importação são definições abstratas que definem um conjunto de operações usando Web Services Description Language (WSDL), uma linguagem XML para descrever serviços da web. Um módulo SCA pode ter muitas interfaces de importação.
- As ligações de importação são definições concretas que especificam o mecanismo físico que os módulos SCA usam para acessar um serviço externo.

Ligações de Importação Suportadas

O IBM Business Process Manager suporta as seguintes ligações de importação:

- As ligações SCA conectam os módulos SCA a outros módulos SCA. As ligações SCA também são referenciadas como ligações padrão.
- As ligações de serviço da web permitem que os componentes chamem serviços da web. Os protocolos suportados são SOAP1.1/HTTP, SOAP1.2/HTTP e SOAP1.1/JMS.

É possível usar uma ligação SOAP1.1/HTTP ou SOAP1.2/HTTP com base em Java API para XML Web Services (JAX-WS), que permite a interação com serviços usando ligações literais de documento ou de RPC e que usa manipuladores JAX-WS para customizar as chamadas. Uma ligação SOAP1.1/HTTP separada é fornecida para permitir interação com serviços que usam uma ligação codificada por RPC ou em que há um requisito para usar manipuladores JAX-RPC para customizar chamadas.

- As ligações HTTP permitem que você acesse aplicativos usando o protocolo HTTP.
- As ligações de importação Enterprise JavaBeans (EJB) permitem que componentes SCA chamem serviços fornecidos pela lógica de negócios Java EE em execução em um servidor Java EE.
- As ligações de Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Essa comunicação é feita por meio do uso de adaptadores de recursos.
- As ligações do Java Message Service (JMS) 1.1 permitem interoperabilidade com o provedor de sistema de mensagem padrão do WebSphere Application Server. O JMS pode explorar vários tipos de transporte, incluindo TCP/IP e HTTP ou HTTPS. A classe de Mensagem JMS e seus cinco subtipos (Texto, Bytes, Objeto, Fluxo e Mapa) são suportados automaticamente.
- As ligações JMS genéricas permitem interoperabilidade com provedores JMS de terceiros que são integrados com o WebSphere Application Server usando o JMS Application Server Facility (ASF).
- As ligações JMS do WebSphere MQ permitem interoperabilidade com os provedores JMS baseados no WebSphere MQ. A classe de Mensagem JMS e seus cinco subtipos (Texto, Bytes, Objeto, Fluxo e Mapa) são suportados automaticamente. Se desejar usar o WebSphere MQ como um provedor JMS, use as ligações JMS do WebSphere MQ.
- As ligações do WebSphere MQ permitem interoperabilidade com o WebSphere MQ. É possível usar as ligações do WebSphere MQ apenas com gerenciadores de fila remotos por meio de uma conexão do cliente do WebSphere MQ; não é possível usá-las com gerenciadores de filas locais. Use as ligações do WebSphere MQ se desejar se comunicar com aplicativos WebSphere MQ nativos.

Chamada Dinâmica de Serviços

Os serviços podem ser chamados por meio de qualquer ligação de importação suportada. Um serviço está normalmente localizado em um terminal especificado na importação. Este terminal é chamado de terminal estático. É possível chamar um serviço diferente substituindo o terminal estático. A substituição dinâmica de terminais estáticos permite chamar um serviço em outro terminal, por meio de qualquer ligação de importação suportada. A chamada dinâmica de serviços também permite que você chame um serviço onde a ligação de importação suportada não tem um terminal estático.

Uma importação com uma ligação associada é usada para especificar o protocolo e sua configuração para chamada dinâmica. A importação usada para a chamada dinâmica pode ser ligada ao componente de chamada ou pode ser selecionada dinamicamente no tempo de execução.

Para chamadas de serviço da web e SCA, também é possível fazer uma chamada dinâmica sem uma importação, com o protocolo e a configuração deduzidos da URL de terminal. O tipo de destino da chamada é identificado a partir da URL do terminal. Se uma importação for usada, a URL deve ser compatível com o protocolo da ligação de importação.

- Uma URL SCA indica a chamada de um outro módulo SCA.
- Um URL de HTTP ou JMS por padrão indica a chamada de um serviço da web; para essas URLs, é possível fornecer um valor de tipo de ligação adicional que indica que a URL representa uma chamada por meio de uma ligação HTTP ou JMS.
- Para uma URL de HTTP de serviço da web, o padrão é usar SOAP 1.1 e pode ser especificado um valor de tipo de ligação que indica o uso de SOAP 1.2.

Exportações e Ligações de Exportações

As exportações definem as interações entre os módulos SCA e os solicitantes de serviços. Os módulos SCA usam exportações para oferecer serviços para outros. As ligações de exportação definem a maneira específica pela qual um módulo SCA é acessado pelos solicitantes de serviço.

Interfaces e Ligações

Uma exportação de módulo SCA precisa de, pelo menos, uma interface.

- Interfaces de exportação são definições abstratas que definem um conjunto de operações usando Web Services Description Language (WSDL), uma linguagem XML para descrever serviços da web. Um módulo SCA pode ter muitas interfaces de exportação.
- As ligações de exportação são definições concretas que especificam o mecanismo físico que os solicitantes de serviço usam para acessar um serviço. Geralmente, uma exportação de módulo SCA possui uma ligação especificada. Uma exportação sem ligação especificada é interpretada pelo ambiente de tempo de execução como uma exportação com uma ligação SCA.

Ligações de Exportação Suportadas

O IBM Business Process Manager suporta as seguintes ligações de exportação:

- As ligações SCA conectam os módulos SCA a outros módulos SCA. As ligações SCA também são referenciadas como ligações padrão.
- As ligações de serviço da web permitem que as exportações sejam chamadas como serviços da web. Os protocolos suportados são SOAP1.1/HTTP, SOAP1.2/HTTP e SOAP1.1/JMS.
É possível usar uma ligação SOAP1.1/HTTP ou SOAP1.2/HTTP com base na Java API for XML Web Services (JAX-WS), que permite interação com os serviços usando ligações do documento ou de literal do RPC e usa os manipuladores JAX-WS para customizar as chamadas. Uma ligação SOAP1.1/HTTP separada é fornecida para permitir interação com serviços que usam uma ligação codificada por RPC ou em que há um requisito para usar manipuladores JAX-RPC para customizar chamadas.
- As ligações HTTP permitem que as exportações sejam acessadas usando o protocolo HTTP.
- As ligações de exportação do Enterprise JavaBeans (EJB) permitem que os componentes do SCA sejam expostos como EJBs para que a lógica de negócios do Java EE possa chamar os componentes do SCA ou, do contrário, indisponibilizá-los.
- As ligações de Enterprise Information System (EIS) fornecem conectividade entre componentes SCA e um EIS externo. Essa comunicação é feita por meio do uso de adaptadores de recursos.
- As ligações do Java Message Service (JMS) 1.1 permitem interoperabilidade com o provedor de sistema de mensagem padrão do WebSphere Application Server. O JMS pode explorar vários tipos de transporte, incluindo TCP/IP e HTTP ou HTTPS. A classe de Mensagem JMS e seus cinco subtipos (Texto, Bytes, Objeto, Fluxo e Mapa) são suportados automaticamente.
- As ligações JMS genéricas permitem interoperabilidade com provedores JMS de terceiros que são integrados com o WebSphere Application Server usando o JMS Application Server Facility (ASF).
- As ligações JMS do WebSphere MQ permitem interoperabilidade com os provedores JMS baseados no WebSphere MQ. A classe de Mensagem JMS e seus cinco subtipos (Texto, Bytes, Objeto, Fluxo e Mapa) são suportados automaticamente. Se desejar usar o WebSphere MQ como um provedor JMS, use as ligações JMS do WebSphere MQ.
- As ligações do WebSphere MQ permitem interoperabilidade com o WebSphere MQ. Use uma conexão remota (ou do cliente) para se conectar com um gerenciador de filas do MQ em uma máquina remota. Uma conexão (ou ligações) local é uma conexão direta com o WebSphere MQ. Isso pode ser usado apenas para uma conexão com um gerenciador de filas do MQ na mesma máquina. O WebSphere MQ permitirá os dois tipos de conexão, mas as ligações do MQ suportarão apenas a conexão "remota" (ou do "cliente").

Módulos de Mediação

Os módulos de mediação são módulos Service Component Architecture (SCA) que podem alterar o formato, o conteúdo ou o destino de solicitações de serviço.

Os módulos de mediação operam em mensagens que estão em andamento entre solicitantes de serviços e provedores de serviços. É possível rotear mensagens para diferentes provedores de serviços e aditar o conteúdo da mensagem ou formulário. Os módulos de mediação podem fornecer funções como criação de log de mensagens e processamento de erro que são customizadas de acordo com seus requisitos.

É possível alterar determinados aspectos de módulos de mediação, a partir do console administrativo, sem precisar reimplementar o módulo.

Componentes dos Módulos de Mediação

Módulos de mediação contêm os seguintes itens:

- Importações, que definem as interações entre os módulos SCA e os provedores de serviços. Elas permitem que os módulos SCA chamem serviços externos como se fossem locais. É possível visualizar as importações do módulo de mediação e modificar a ligação.
- Exportações, que definem as interações entre os módulos SCA e os solicitantes de serviços. Elas permitem que um módulo SCA ofereça um serviço e defina as interfaces externas (pontos de acesso) de um módulo SCA. É possível visualizar as exportações do módulo de mediação.
- Componentes SCA, que são blocos de construção para módulos SCA, tais como módulos de mediação. É possível criar e customizar módulos SCA e componentes graficamente, usando Integration Designer. Depois de implementar um módulo de mediação, é possível customizar determinados aspectos dele a partir do console administrativo, sem precisar reimplementar o módulo.

Geralmente, os módulos de mediação contêm um tipo específico de componente SCA chamado de *componente de fluxo de mediação*. Os componentes do fluxo de mediação definem fluxos de mediação.

Um componente do fluxo de mediação pode conter nenhuma, uma ou várias primitivas de mediação. O IBM Business Process Manager suporta um conjunto fornecido de primitivas de mediação que fornecem funcionalidade para roteamento e transformação de mensagem. Para flexibilidade adicional da primitiva de mediação, use a Primitiva de mediação customizada para chamar a lógica customizada.

O propósito de um módulo de mediação que não contém um componente do fluxo de mediação é transformar solicitações de serviço de um protocolo para outro. Por exemplo, um pedido de serviço pode ser feito utilizando SOAP/JMS mas pode precisar ser transformado para SOAP/HTTP antes do envio.

Nota: É possível visualizar e fazer determinadas alterações nos módulos de mediação a partir do IBM Business Process Manager. No entanto, não é possível visualizar ou alterar os componentes SCA dentro de um módulo a partir do IBM Business Process Manager. Use o Integration Designer para customizar componentes SCA.

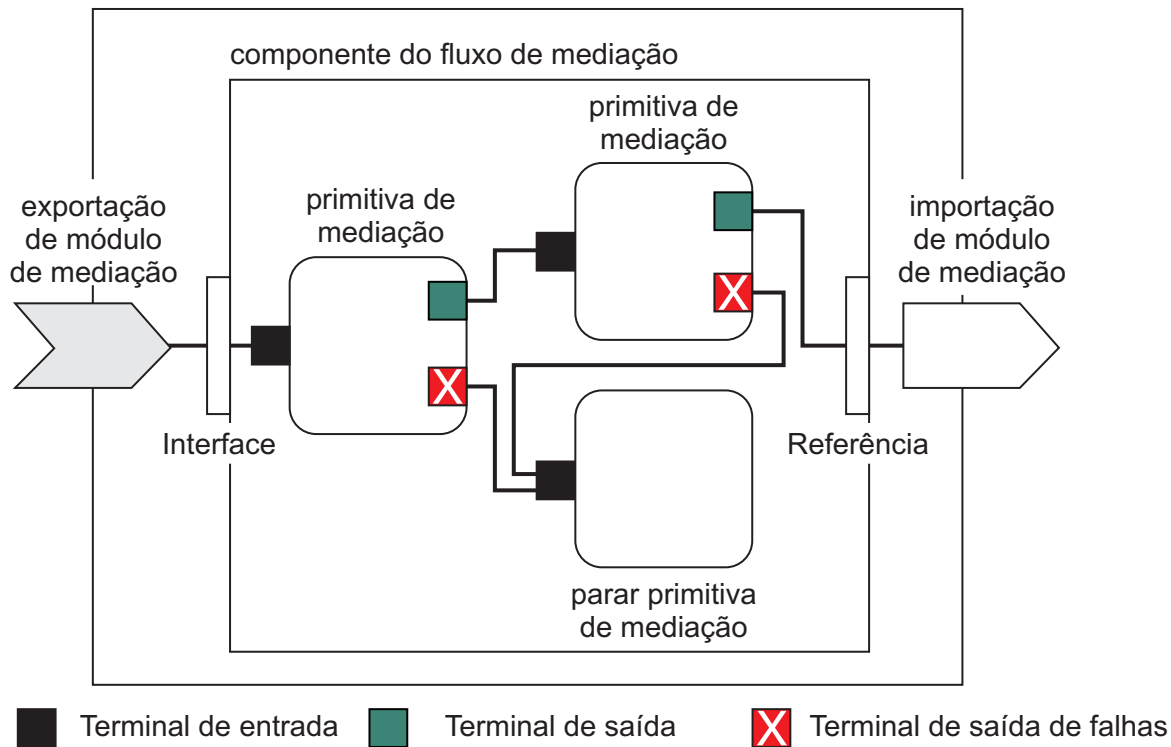


Figura 88. Exemplo simplificado de um módulo de mediação. O módulo de mediação contém um componente do fluxo de mediação, que contém primitivas de mediação.

- Propriedades

As primitivas de mediação possuem propriedades, algumas das quais podem ser exibidas no console administrativo como propriedades adicionais de um módulo SCA.

Para propriedades da primitiva de mediação ficarem visíveis a partir do console administrativo do IBM Business Process Manager, o desenvolvedor de integração deve promover as propriedades.

Determinadas propriedades aperfeiçoam-se sendo administrativamente configuradas e o Integration Designer as descreve como propriedades promovíveis, pois elas podem ser promovidas a partir do ciclo de integração para o ciclo administrativo. Outras propriedades não são adequadas para configuração administrativa, porque a modificação delas pode afetar o fluxo de mediação de tal maneira que o módulo de mediação precisa ser reimplementado. O Integration Designer lista as propriedades que você pode escolher para promover nas propriedade promovidas de uma primitiva de mediação.

É possível usar o console administrativo do IBM Business Process Manager para alterar o valor das propriedades promovidas sem precisar reimplementar um módulo de mediação ou reiniciar o servidor ou módulo.

Geralmente, os fluxos de mediação usam as alterações de propriedade imediatamente. No entanto, se as mudanças de propriedade ocorrerem em uma célula do gerenciador de implementação, elas terão efeito em cada nó conforme esse nó for sincronizado. Além disso, os fluxos de mediação que estão em andamento continuam utilizando valores anteriores.

Nota: No console administrativo, você só pode alterar os valores da propriedade, não grupos de propriedades, nomes ou tipos. Se quiser alterar grupos de propriedades, nomes ou tipos, você deve usar o Integration Designer.

- Um módulo de mediação ou biblioteca dependente também pode definir os subfluxos. Um subfluxo contém um conjunto de primitivas de mediação ligadas como uma parte reutilizável da lógica de integração. Uma primitiva pode ser incluída em um fluxo de mediação para chamar um subfluxo.

Implementando Módulos de Mediação

Os módulos de mediação são criados usando o Integration Designer e são geralmente implementados no IBM Business Process Manager dentro de um arquivo de archive corporativo (EAR).

Você pode alterar o valor das propriedades promovidas no momento da implementação.

É possível exportar um módulo de mediação a partir do Integration Designer e fazer com que o Integration Designer empacote o módulo de mediação dentro de um arquivo Java archive (JAR) e o arquivo JAR dentro de um arquivo EAR. Você pode, então, implementar o arquivo EAR, instalando um novo aplicativo a partir do console administrativo.

Os módulos de mediação podem ser considerado como uma entidade. No entanto, os módulos SCA são definidos por vários arquivos XML armazenados em um arquivo JAR.

Exemplo de arquivo EAR, contendo um módulo de mediação

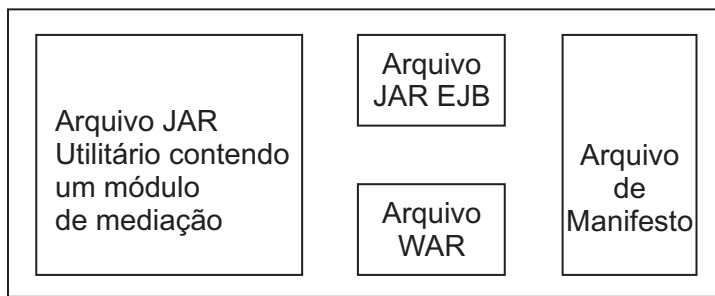


Figura 89. Exemplo simplificado de um arquivo EAR contendo um módulo de mediação. O arquivo EAR contém JARs. O arquivo JAR utilitário contém um módulo de mediação.

Primitivas de Mediação

Os componentes do fluxo de mediação operam em fluxos de mensagens entre componentes de serviço. Os recursos de um componente de mediação são implementados pelas *primitivas de mediação*, que implementam tipos de implementação de serviço padrão.

Um componente do fluxo de mediação possui um ou mais fluxos. Por exemplo, um para pedido e um para resposta.

O IBM Business Process Manager suporta um conjunto fornecido de primitivas de mediação, o qual implementa recursos de mediação padrão para módulos de mediação ou módulos implementados no IBM Business Process Manager. Se precisar de recursos de mediação especiais, será possível desenvolver suas próprias primitivas de mediação customizadas.

Uma primitiva de mediação define uma operação de “entrada” que processa ou manipula mensagens que são representadas por SMOs (Service Message Objects). Uma primitiva de mediação também pode definir operações de “saída” que enviam mensagens para outro componente ou módulo.

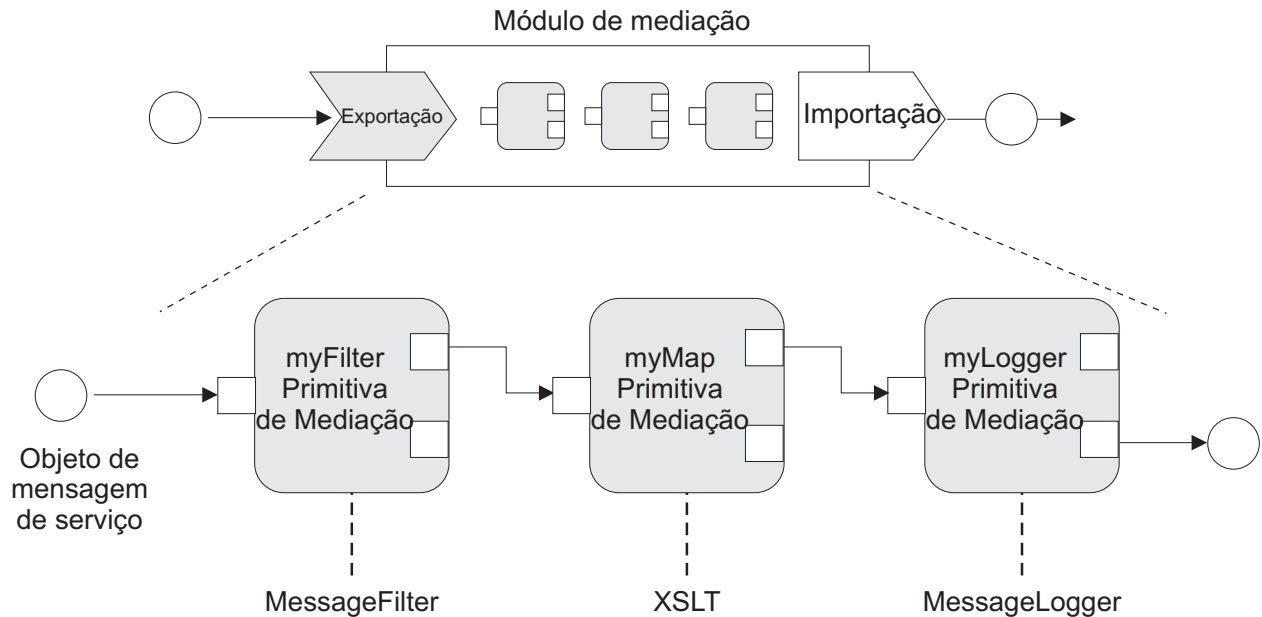


Figura 90. Módulo de Mediação Contendo Três Primitivas de Mediação

Você pode utilizar o Integration Designer para configurar primitivas de mediação e configurar suas propriedades. Algumas destas propriedades podem ficar visíveis para o administrador de tempo de execução sendo promovidas. Qualquer propriedade de primitiva de mediação que pode ser promovida também pode ser uma propriedade dinâmica. Uma propriedade dinâmica pode ser substituída, no tempo de execução, utilizando um arquivo de políticas.

O Integration Designer também permite modelar e montar graficamente componentes do fluxo de mediação a partir de primitivas de mediação e montar módulos de mediação ou módulos a partir dos componentes do fluxo de mediação. O console administrativo refere-se a módulos de mediação e módulos como módulos SCA.

Integration Designer também permite a definição de subfluxos nos módulos ou em suas bibliotecas dependentes. Um subfluxo pode conter qualquer primitiva de mediação, exceto para a primitiva de mediação Resolução de Política. Um subfluxo é chamado a partir de um fluxo de pedido ou resposta ou de outro subfluxo utilizando a primitiva de mediação Subfluxo. As propriedades promovidas a partir das primitivas de mediação em um subfluxo são expostas como propriedades nas primitivas de mediação Subfluxo. Elas podem, então, ser promovidas novamente até que alcancem o nível do módulo; nesse ponto elas poderão ser modificadas pelo administrador de tempo de execução.

Primitivas de Mediação Suportadas

O conjunto de primitivas de mediação a seguir é suportado pelo IBM Business Process Manager:

Mapa de Objetos de Negócios

Transforma mensagens.

- Define transformações de mensagens utilizando um mapa de objetos de negócios, que pode ser reutilizado.
- Permite definir transformações de mensagens graficamente, utilizando o editor de mapas de objetos de negócios.
- Pode alterar o conteúdo de uma mensagem.
- Pode transformar um tipo de mensagem de entrada em um tipo de mensagem de saída diferente.

Mediação Customizada

Permite implementar sua própria lógica de mediação no código Java. A primitiva de Mediação Customizada combina a flexibilidade de uma primitiva de mediação definida pelo usuário, com a simplicidade de uma primitiva de mediação predefinida. É possível criar transformações complexas e padrões de roteamento:

- Criando código Java.
- Criando suas próprias propriedades.
- Incluindo novos terminais.

É possível chamar um serviço a partir de uma primitiva de Mediação Customizada, mas a primitiva de mediação Chamada de Serviço foi projetada para chamar serviços e fornece funcionalidade adicional, como uma nova tentativa.

Manipulador de Dados

Permite transformar uma parte de uma mensagem. É utilizado para converter um elemento de uma mensagem de um formato físico em uma estrutura lógica ou de uma estrutura lógica em um formato físico. O principal uso da primitiva é converter um formato físico, como uma cadeia de Texto em um objeto de Mensagem de Texto JMS, em uma estrutura lógica do Objeto de Negócios e fazer a conversão de volta. Esta mediação é comumente utilizada para:

- Transforme uma seção da mensagem de entrada de uma estrutura definida em outra - um exemplo seria quando o SMO inclui um valor de sequência que é delimitado por vírgula e deseja analisar para um Objeto de Negócios específico.
- Alterar o tipo de mensagem – um exemplo disto seria quando uma exportação JMS tiver sido configurada para utilizar uma ligação de dados básica categorizada por JMS e, no módulo de mediação, o desenvolvedor de integração decide se o conteúdo deve ser expandido para uma estrutura de BO específica.

Consulta ao Banco de Dados

Modifica mensagens, utilizando informações de um banco de dados fornecido pelo usuário.

- Você deve configurar um banco de dados, uma origem de dados e quaisquer configurações de autenticação de servidor para a primitiva de mediação de Consulta do Banco de Dados usar. Utilize o console administrativo para ajudá-lo a fazer isso.
- A primitiva de mediação de Consulta do Banco de Dados pode ler a partir apenas de uma tabela.
- A coluna-chave especificada deve conter um valor exclusivo.
- Os dados nas colunas de valor devem ter um tipo de esquema XML simples ou um tipo de esquema XML que estende um tipo de esquema XML simples.

Consulta do Terminal

Permite o roteamento dinâmico de pedidos, procurando terminais de serviço em um repositório.

- As informações do terminal em serviço são recuperadas de um WebSphere Service Registry and Repository (WSRR). O registro WSRR pode ser local ou remoto.
- Você faz alterações no registro a partir do console administrativo do WSRR.
- O IBM Business Process Manager precisa saber qual registro utilizar, portanto, é necessário criar definições de acesso de WSRR utilizando o console administrativo do IBM Business Process Manager.

Emissor de Eventos

Melhora o monitoramento permitindo o envio de eventos de dentro de um componente do fluxo de mediação.

- É possível suspender a ação de mediação ao desmarcar a caixa de opção.
- É possível visualizar os eventos do Emissor de Eventos usando o navegador do Commom Base Events no IBM Business Process Manager.
- Você deve enviar apenas eventos em um ponto significativo em um fluxo de mediação, por motivos de desempenho.

- É possível definir as partes da mensagem contidas no evento.
- Os eventos são enviados em forma de Common Base Events e são enviados para um servidor Common Event Infrastructure.
- Para utilizar totalmente as informações do Emissor de Eventos, os consumidores de eventos precisam entender a estrutura dos Common Base Events. Os Common Base Events possuem um esquema geral, mas isto não modela os dados específicos do aplicativo, que estão contidos nos elementos de dados estendidos. Para modelar os elementos de dados estendidos, as ferramentas do Integration Designer geram um arquivo de definição do catálogo de eventos de Common Event Infrastructure para cada uma das primitivas de mediação do Emissor de Evento configurado. Os arquivos de definição de catálogo de eventos são artefatos de exportação fornecidos para ajudá-lo; eles não são usados pelo Integration Designer ou pelo tempo de execução do IBM Business Process Manager. Você deve consultar os arquivos de definição do catálogo de eventos ao criar aplicativos para consumir eventos do Emissor de Eventos.
- Você pode especificar outro monitoramento no IBM Business Process Manager. Por exemplo, você pode monitorar eventos a serem emitidos a partir de importações e exportações.

Falha Pára um caminho específico no fluxo e gera uma exceção.

Fan In Ajuda a agregar (combinar) mensagens.

- Pode ser utilizado apenas em combinação com a primitiva de mediação Fan Out.
- Juntas, as primitivas de mediação Fan Out e Fan In permitem a agregação de dados em uma mensagem de saída.
- A primitiva de mediação Fan In recebe mensagens até que um ponto de decisão seja alcançado, em seguida, ocorre uma saída de mensagem.
- O contexto compartilhado deve ser utilizado para conter dados de agregação.

Fan Out

Ajuda a dividir e agregar (combinar) mensagens.

- Juntas, as primitivas de mediação Fan Out e Fan In permitem a agregação de dados em uma mensagem de saída.
- No modo de iteração, a primitiva de mediação Fan Out permite iterar por meio de uma única mensagem de entrada que contém o elemento de repetição. Para cada ocorrência do elemento de repetição, é enviada uma mensagem.
- O contexto compartilhado deve ser utilizado para conter dados de agregação.

Configurador do Cabeçalho HTTP

Fornece um mecanismo para gerenciar cabeçalhos em mensagens HTTP.

- Pode criar, configurar, copiar ou excluir cabeçalhos da mensagem HTTP.
- Pode configurar várias ações para alterar vários cabeçalhos HTTP.

Mapeamento

Transforma mensagens.

- Permite executar transformações Linguagem de Folha de Estilo Extensível (XSL) ou de Mapa de Objetos de Negócios.
- Você transforma mensagens usando uma transformação XSLT 1.0 ou XSLT 2.0 ou Mapa de Objeto de Negócios. As transformações XSL operam em uma serialização XML da mensagem, enquanto que a transformação Mapa de Objeto de Negócios opera no Service Data Objects (SDO).

Configurador do Elemento de Mensagem

Fornece um mecanismo simples de configurar o conteúdo de mensagens.

- Pode alterar, incluir ou excluir elementos de mensagem.
- Não altera o tipo da mensagem.

- Os dados nas colunas de valor devem ter um tipo de esquema XML simples ou um tipo de esquema XML que estende um tipo de esquema XML simples.

Filtro de Mensagem

Roteia mensagens por diferentes caminhos, com base no conteúdo da mensagem.

- É possível suspender a ação de mediação ao desmarcar a caixa de opção.

Registrador de Mensagem

Registra mensagens em um banco de dados relacional ou por meio de seu próprio criador de log. As mensagens são armazenadas como XML, portanto, os dados podem ser pós-processados por aplicativos identificados por XML.

- É possível suspender a ação de mediação ao desmarcar a caixa de opção.
- O esquema do banco de dados relacional (estrutura de tabela) é definido pela IBM.
- Por padrão, a primitiva de mediação Message Logger utiliza o banco de dados Comum. O ambiente de tempo de execução mapeia a origem de dados em `jdbc/mediation/messageLog` para o banco de dados Comum.
- Você pode configurar classes de implementação do Manipulador para customizar o comportamento do criador de log customizado. Opcionalmente, é possível fornecer classes de implementação do Formatador, classes de implementação do Filtro ou ambas para customizar o comportamento do criador de log customizado.

Configurador do Cabeçalho do MQ

Fornece um mecanismo para gerenciar cabeçalhos em mensagens do MQ.

- Pode criar, configurar, copiar ou excluir cabeçalhos da mensagem do MQ.
- Pode configurar várias ações para alterar vários cabeçalhos do MQ.

Resolução de Política

Permite a configuração dinâmica de pedidos, procurando terminais em serviço e arquivos de políticas associados, em um repositório.

- É possível utilizar um arquivo de políticas para substituir dinamicamente as propriedades promovidas de outras primitivas de mediação.
- As informações do terminal em serviço e as informações de política são recuperadas de um WebSphere Service Registry and Repository (WSRR). O registro WSRR pode ser local ou remoto.
- Você faz alterações no registro a partir do console administrativo do WSRR.
- O IBM Business Process Manager precisa saber qual registro utilizar, portanto, é necessário criar definições de acesso de WSRR utilizando o console administrativo do IBM Business Process Manager.

Chamada de Serviço

Chama um serviço a partir de um fluxo de mediação em vez de aguardar até o final do fluxo de mediação e utiliza o mecanismo callout.

- Se o serviço retornar uma falha, você poderá tentar novamente o mesmo serviço ou chamar outro serviço.
- A primitiva de mediação Chamada de Serviço é uma poderosa primitiva de mediação que pode ser utilizada para chamadas de serviço simples, ou em combinação com outras primitivas de mediação para mediações complexas.

Tipo de Mensagem Configurado

Durante o desenvolvimento de integração, permite tratar campos de mensagens de texto simples como se fossem de texto restrito. Um campo é de texto simples se puder conter mais de um tipo de dados. Um campo é de texto restrito se seu tipo de estrutura interna forem conhecidos.

- No tempo de execução, a primitiva de mediação Tipo de Mensagens permite verificar se o conteúdo de uma mensagem corresponde aos tipos de dados esperados.

Configurador do Cabeçalho SOAP

Fornecer um mecanismo para gerenciar cabeçalhos em mensagens SOAP.

- Pode criar, configurar, copiar ou excluir cabeçalhos da mensagem SOAP.
- Pode configurar várias ações para alterar vários cabeçalhos SOAP.

Parar Pára um caminho específico no fluxo, sem gerar uma exceção.

Filtro de Tipo

Permite direcionar mensagens para um caminho diferente de um fluxo, com base em seu tipo.

WebSphere eXtreme Scale Recuperar

É possível recuperar informações de um ambiente de cache do servidor do eXtreme Scale.

- É possível consultar valores no cache e armazená-los como elementos na mensagem usando uma chave.
- Combinando as primitivas de mediação eXtreme Scale Armazenar e Recuperar, é possível armazenar em cache a resposta de um sistema backend. Solicitações futuras não exigirão acesso a esse sistema backend.
- Você deve criar definições do eXtreme Scale usando o console administrativo WebSphere ESB, portanto será possível especificar qual servidor eXtreme Scale usar.

WebSphere eXtreme Scale Armazenar

É possível armazenar informações em um ambiente de cache do servidor eXtreme Scale.

- É possível armazenar informações em um cache do eXtreme Scale usando uma chave e um objeto.
- Combinando as primitivas de mediação eXtreme Scale Armazenar e Recuperar, é possível usar a primitiva de mediação Armazenar para armazenar dados no cache e usar a primitiva de mediação Recuperar para recuperar dados armazenados anteriormente na cache.
- Você deve criar definições do eXtreme Scale usando o console administrativo WebSphere ESB, portanto será possível especificar qual servidor eXtreme Scale usar.

Roteamento Dinâmico

É possível rotear mensagens de várias maneiras usando terminais definidos no momento da integração ou terminais determinados, dinamicamente, no tempo de execução.

Roteamento dinâmico abrange dois casos de roteamento de mensagem:

- Roteamento de mensagem em que o fluxo é dinâmico, mas todos os terminais possíveis são predefinidos em um módulo de Service Component Architecture (SCA).
- Roteamento de mensagem em que o fluxo é dinâmico e a seleção de terminal também é dinâmica. Os terminais de serviço são selecionados de uma origem externa no tempo de execução.

Seleção de Terminal Dinâmico

O tempo de execução tem a capacidade de rotear as mensagens de solicitação e resposta para um endereço de terminal identificado por um elemento de cabeçalho da mensagem. Este elemento de cabeçalho da mensagem pode ser atualizado pelas primitivas de mediação, em um fluxo de mediação. O endereço de terminal poderá ser atualizado com as informações de um registro, um banco de dados ou com informações da própria mensagem. O roteamento de mensagens de resposta se aplica apenas quando a resposta está sendo enviada por uma exportação JAX-WS de serviço da web.

Para que o tempo de execução implemente o roteamento dinâmico em uma solicitação ou resposta, o módulo SCA deve ter a propriedade Usar terminal dinâmico se configurado no cabeçalho da mensagem configurada. Os desenvolvedores de integração podem configurar a propriedade Usar terminal dinâmico se configurado no cabeçalho da mensagem ou poderão promovê-la (torná-la visível no tempo de execução), para que o administrador de tempo de execução possa configurá-la. É possível visualizar as propriedades do módulo na janela Propriedades do Módulo. Para ver a janela, clique em **Aplicativos > Módulos SCA > Propriedades do Módulo**. O desenvolvedor de integração fornece nomes de alias de

propriedades promovidas e esses são os nomes exibidos no console administrativo.

Registro

É possível usar o IBM WebSphere Service Registry and Repository (WSRR) para armazenar informações do terminal em serviço e, em seguida, criar módulos SCA para recuperar terminais a partir do registro do WSRR.

Quando desenvolver módulos SCA, será possível usar a primitiva de mediação de Consulta de Terminal para permitir que um fluxo de mediação consulte um registro do WSRR para um terminal em serviço ou um conjunto de terminais em serviço. Se um módulo SCA recuperar um conjunto de terminais, ele deverá usar uma outra primitiva de mediação para selecionar o preferencial.

Controle de Política de Mediação de Solicitações de Serviço

É possível usar políticas de mediação para controlar fluxos de mediação entre solicitantes de serviços e provedores de serviços.

É possível controlar os fluxos de mediação usando políticas de mediação armazenadas no IBM WebSphere Service Registry and Repository (WSRR). A implementação de gerenciamento de política de serviço no WSRR é baseada no Web Services Policy Framework (WS-Policy).

Para controlar as solicitações de serviço usando políticas de mediação, você deve ter módulos adequados de Service Component Architecture (SCA) e documentos de política de mediação em seu registro do WSRR.

Como Conectar uma Política de Mediação a uma Solicitação de Serviço

Ao desenvolver um módulo SCA que precisar fazer uso de uma política de mediação, você deve incluir uma primitiva de mediação de Resolução de Política no fluxo de mediação. No tempo de execução, a primitiva de mediação de Resolução de Política obtém informações da política de mediação a partir do registro. Portanto, um módulo SCA deve conter um componente do fluxo de mediação para suportar o controle de solicitações de serviço da política de mediação.

No registro, é possível anexar uma ou mais políticas de mediação para um módulo SCA ou para um serviço de destino usado pelo módulo SCA. As políticas de mediação anexadas podem ser usadas (estão no escopo) para todas as mensagens de serviço processadas por esse módulo SCA. As políticas de mediação podem ter anexos de política que definem as condições. As condições de política de mediação permitem que políticas de mediação diferentes sejam aplicadas em contextos diferentes. Além disso, as políticas de mediação podem ter classificações, que podem ser usadas para especificar um estado de controle.

WebSphere Service Registry and Repository

O produto WebSphere Service Registry and Repository (WSRR) permite armazenar, acessar e gerenciar informações sobre terminais de serviço e políticas de mediação. É possível usar o WSRR para tornar seus aplicativos de serviço mais dinâmicos e mais adaptáveis para alterar as condições de negócios.

Introdução

Os fluxos de mediação podem usar WSRR como um mecanismo de consulta dinâmico, fornecendo informações sobre terminais de serviço ou políticas de mediação.

Para configurar o acesso ao WSRR, você cria documentos de definição do WSRR utilizando o console administrativo. Uma alternativa é utilizar os comandos de administração do WSRR do cliente de script wsadmin. As definições de WSRR e suas propriedades da conexão são o mecanismo usado para se conectar a uma instância de registro e recuperar um terminal em serviço ou uma política de mediação.

Terminais em Serviço

É possível utilizar o WSRR para armazenar informações sobre serviços que já utiliza, que planeja utilizar ou sobre os quais gostaria de estar ciente. Estes serviços podem estar em seus ou em outros sistemas. Por exemplo, um aplicativo poderia usar o WSRR para localizar o serviço mais apropriado para satisfazer suas necessidades funcionais e de desempenho.

Ao desenvolver um módulo SCA que precisa acessar terminais em serviço a partir do WSRR, você deve incluir uma primitiva de mediação de Consulta de Terminal no fluxo de mediação. No tempo de execução, a primitiva de mediação Consulta de Terminal obtém os terminais em serviço do registro.

Políticas de Mediação

Também é possível usar o WSRR para armazenar as informações sobre política de mediação. As políticas de mediação podem ajudar a controlar as solicitações de serviço, substituindo dinamicamente as propriedades do módulo. Se o WSRR contiver políticas de mediação que são anexadas a um objeto representando seu módulo SCA ou seu serviço de destino, as políticas de mediação poderão substituir as propriedades do módulo. Se desejar que políticas de mediação diferentes sejam aplicadas em diferentes contextos, é possível criar condições de política de mediação.

Nota: As políticas de mediação estão preocupadas com o controle dos fluxos de mediação e não com a segurança.

Ao desenvolver um módulo SCA que precisar fazer uso de uma política de mediação, você deve incluir uma primitiva de mediação de Resolução de Política no fluxo de mediação. No tempo de execução, a primitiva de mediação de Resolução de Política obtém informações da política de mediação a partir do registro.

WebSphere eXtreme Scale

Ao usar o produto WebSphere eXtreme Scale (eXtreme Scale), é possível fornecer um sistema de armazenamento em cache que pode ser integrado a um aplicativo IBM Business Process Manager. Usando o eXtreme Scale com IBM Business Process Manager é possível melhorar os tempos de resposta de serviço e a confiabilidade, e fornecer uma funcionalidade de integração adicional.

eXtreme Scale atua como uma grade de dados elásticos, escaláveis e de memória interna. A grade de dados armazena em cache, particiona, replica e gerencia dinamicamente os dados do aplicativo e a lógica de negócios em diversos servidores. Com o eXtreme Scale, também é possível obter as qualidades do serviço como integridade transacional, alta disponibilidade e tempos de resposta previsíveis.

É possível usar fluxos de mediação para acessar a função de armazenamento em cache do eXtreme Scale incluindo as primitivas de mediação do WebSphere eXtreme Scale em seu fluxo. Ao desenvolver um módulo Service Component Architecture (SCA) que precisa armazenar informações em um cache do eXtreme Scale, você deve incluir a primitiva de mediação Armazenar do WebSphere eXtreme Scale no fluxo de mediação. Se deseja recuperar informações de um cache do eXtreme Scale, você deve incluir a primitiva de mediação Recuperar do WebSphere eXtreme Scale. Ao combinar as duas primitivas de mediação em um fluxo de mediação, é possível armazenar em cache a resposta de um sistema backend, para que as solicitações futuras possam recuperar a resposta do cache.

Para configurar o acesso ao eXtreme Scale, você deve criar uma definição do WebSphere eXtreme Scale usando o console administrativo. Como alternativa, é possível usar os comandos de administração do WebSphere eXtreme Scale a partir do cliente de script wsadmin. Uma definição do eXtreme Scale é o mecanismo usado pelas primitivas de mediação Recuperar e Armazenar do WebSphere eXtreme Scale para se conectar a um servidor eXtreme Scale.

Clientes do Serviço de Mensagens

Os clientes do serviço de mensagens estão disponíveis para C/C++ e .NET para permitir que aplicativos não Java se conectem ao barramento de serviço corporativo.

O Message Service Clients for C/C++ and .NET fornece uma API chamada XMS que possui o mesmo conjunto de interfaces que a API do Sistema de Mensagens Java (JMS). O Cliente do Serviço de Mensagens para C/C++ contém duas implementações do XMS, uma para uso pelos aplicativos C e outra para uso pelos aplicativos C++. O Message Service Client for .NET contém uma implementação totalmente gerenciada do XMS, que pode ser usada por qualquer linguagem compatível com .NET.

É possível obter os Clientes do Serviço de Mensagens para .NET a partir do http://www-01.ibm.com/support/docview.wss?rs=0&q1=IA9H&uid=swg24011756&loc=en_US&cs=utf-8&cc=us&lang=en

É possível obter Clientes do Serviço de Mensagens para C/C++ a partir do http://www-01.ibm.com/support/docview.wss?rs=0&q1=ia94&uid=swg24007092&loc=en_US&cs=utf-8&cc=us&lang=en.

Também é possível instalar e usar o suporte a clientes Java EE do WebSphere Application Server Network Deployment, incluindo cliente de serviços da web, Clientes EJB e Cliente JMS.

