

IBM Business Process Manager
Wersja 8 Wydanie 5

*Przegląd produktu IBM Business
Process Manager*

IBM

Podręczniki w formacie PDF oraz Centrum informacyjne

Podręczniki w formacie PDF zostały udostępnione w celu ułatwienia drukowania i czytania w trybie bez połączenia. Aby uzyskać najnowsze informacje, należy korzystać z Centrum informacyjnego dostępnego w trybie z połączeniem.

Podręczniki w formacie PDF zawierają tę samą treść co Centrum informacyjne (jako zestaw). Niektóre odsyłacze opublikowane w podręcznikach w formacie PDF zostały dostosowane do użycia w Centrach informacyjnych i mogą nie działać poprawnie.

Dokumentacja w formacie PDF jest dostępna w przeciągu kwartału od opublikowania wersji głównej Centrum informacyjnego, np. wersji 7.0 lub wersji 7.5.

Dokumentacja w formacie PDF jest aktualizowana rzadziej niż Centrum informacyjne, ale częściej niż dokumentacja techniczna (Redbooks). Podręczniki w formacie PDF są na ogół aktualizowane, kiedy zgromadzono odpowiednią liczbę zmian, aby mogły one zostać wprowadzone w podręczniku.

Spis treści

**Podręczniki w formacie PDF oraz
Centrum informacyjne iii**

Rozdział 1. Pierwsze kroki w programie IBM Business Process Manager 1

Przegląd produktu	1
Konfiguracje produktu IBM Business Process Manager	3
Możliwości konfiguracji produktu IBM Business Process Manager	4
Repozytorium komponentu Process Center	4
Komponent Process Server a środowiska wykonawcze	6
Środowiska tworzenia treści	6
Narzędzia administracyjne.	8
Ułatwienia dostępu w produkcie IBM Business Process Manager	9
Dostępność języków narodowych w produkcie IBM Business Process Manager	10
Przegląd zarządzania procesami biznesowymi	11
Przegląd modelowania procesów	12
Programowanie procesów z użyciem komponentu Process Center	13
Aplikacje procesów: przegląd	14
Uruchamianie i debugowanie procesów	19
Instalowanie aplikacji procesów i zarządzanie nimi	20
Tworzenie usług, dostęp do usług oraz ich włączanie	22
Uzyskiwanie dostępu do usług zewnętrznych dla aplikacji.	22
Tworzenie lub wywoływanie usługi Web Service	27
Dodatkowe informacje o kluczowych pojęciach	29
Kontrola wersji.	29
Kontrola wersji aplikacji procesów	29
Kontrola wersji modułów i bibliotek	30
Moduły i biblioteki powiązane z aplikacjami procesów lub bibliotekami narzędziowymi	31
Konwencje nazewnictwa	31
Konwencje nazewnictwa dla wdrożeń serwera komponentu Process Center	32
Konwencje nazewnictwa dla wdrożeń na serwerze Process Server	35
Powiązania z uwzględnieniem wersji	36
Wywoływanie dynamiczne z uwzględnieniem wersji	38
Wdrażanie aplikacji procesów z modułami i projektami Java	38
Wdrażanie aplikacji procesów z regułami biznesowymi i selektorami	38
Obiekty konfiguracyjne	39
Architektura wdrażania	39
Komórki	39
Serwery	39
Serwery autonomiczne	40
Klastry	40
Profile	41
Menedżery wdrażania.	42
Węzły	42
Węzły zarządzane	43

Węzły niezarządzane	43
Agenty węzłów	43
Uwagi na temat nazewnictwa profili, węzłów, serwerów, hostów i komórek.	43
Specyfikacja BPMN 2.0	47
Definicje procesów biznesowych (BPD)	50
Powiązania	50
Przegląd powiązań eksportów i importów	52
Konfiguracja powiązania eksportu i importu	56
Transformacja formatu danych w importach i eksportach	56
Selektory funkcji w powiązaniach eksportu	60
Obsługa błędów	62
Współdziałanie między modułami SCA i usługami Open SCA	67
Typy powiązań.	69
Wybór odpowiednich powiązań.	69
Powiązania SCA	71
Powiązania usług Web Services.	71
Powiązania HTTP	96
Powiązania EJB	103
Powiązania EIS	110
Powiązania JMS	116
Ogólne powiązania JMS	124
Powiązania WebSphere MQ JMS	131
Powiązania produktu WebSphere MQ	138
Ograniczenia powiązań	148
Obiekty biznesowe	149
Definiowanie obiektów biznesowych	150
Praca z obiektami biznesowymi	151
Specjalne obiekty biznesowe	152
Tryb analizowania obiektu biznesowego	153
Uwagi na temat wybierania trybu analizowania obiektu biznesowego	153
Zalety stosowania opóźnionego (zamiast przyspieszonego) trybu analizowania	154
Uwagi na temat migrowania i programowania aplikacji	155
Relacje.	156
Usługa relacji	159
Menedżer relacji	159
Relacje w środowiskach wdrożenia sieciowego	160
Funkcje API usługi relacji	160
Magistrala usług korporacyjnych w produkcie IBM Business Process Manager	160
Łączenie usług za pośrednictwem magistrali usług korporacyjnych	160
Infrastruktura przesyłania komunikatów magistrali usług korporacyjnych	162
Hosty przesyłania komunikatów lub miejsca docelowego kolejki	162
Dostawcy JDBC	163
Magistrala integracji usług dla produktu IBM Business Process Manager	163
Aplikacje usług i moduły usług	163
Importy i powiązania importu	164

Eksperty i powiązania eksportów	166
Moduły mediacji	167
Operacje podstawowe mediacji	169
Dynamiczne kierowanie.	174
Sterowanie żadaniami usług przy użyciu strategii mediacji.	175
WebSphere Service Registry and Repository	175
WebSphere eXtreme Scale	176
Klienci usługi komunikatów	176

Rozdział 2. Dodatkowe informacje o kluczowych pojęciach. 179

Kontrola wersji	179
Kontrola wersji aplikacji procesów	179
Kontrola wersji modułów i bibliotek	180
Moduły i biblioteki powiązane z aplikacjami procesów lub bibliotekami narzędziowymi	181
Konwencje nazewnictwa	181
Konwencje nazewnictwa dla wdrożeń serwera komponentu Process Center	182
Konwencje nazewnictwa dla wdrożeń na serwerze Process Server	185
Powiązania z uwzględnianiem wersji	186
Wywoływanie dynamiczne z uwzględnianiem wersji	188
Wdrażanie aplikacji procesów z modułami i projektami Java	188
Wdrażanie aplikacji procesów z regułami biznesowymi i selektorami	188
Obiekty konfiguracyjne	189
Architektura wdrażania	189
Komórki	189
Serwery	189
Serwery autonomiczne	190
Klastry	190
Profile	191
Menedżery wdrażania	192
Węzły	192
Węzły zarządzane	193
Węzły niezarządzane	193
Agenty węzłów	193
Uwagi na temat nazewnictwa profili, węzłów, serwerów, hostów i komórek	193
Specyfikacja BPMN 2.0.	197
Definicje procesów biznesowych (BPD)	200
Powiązania	200
Przegląd powiązań eksportów i importów	202
Konfiguracja powiązania eksportu i importu.	206
Transformacja formatu danych w importach i eksportach.	206
Procedury obsługi danych	207
Powiązania danych	209
Selektory funkcji w powiązaniach eksportu	210
Obsługa błędów	212
Sposób obsługi błędów w powiązaniach eksportu	213
Sposób obsługi błędów w powiązaniach importu	215
Współdziałanie między modułami SCA i usługami	
Open SCA.	217
Typy powiązań	219
Wybór odpowiednich powiązań	219

Powiązania SCA	221
Powiązania usług Web Services	221
Przegląd powiązań usług Web Services	221
Propagowanie nagłówków SOAP	222
Propagowanie nagłówków transportu.	225
Praca z powiązaniem usługi Web Services	
JAX-WS	227
Załączniki w komunikatach SOAP	230
Używanie powiązania stylu dokumentu WSDL z komunikatami wieloczęściowymi	244
Powiązania HTTP	246
Przegląd powiązań HTTP	246
Nagłówki HTTP	247
Powiązania danych HTTP	251
Powiązania EJB	253
Powiązania importu EJB	254
Powiązania eksportu EJB	255
Właściwości powiązania EJB	256
Powiązania EIS	260
Przegląd powiązań EIS	260
Kluczowe opcje powiązań z systemami EIS	261
Właściwości dynamiczne JCA specyfikacji interakcji i specyfikacji połączenia	263
Klienci zewnętrzne z powiązaniem EIS	265
Powiązania JMS	265
Przegląd powiązań JMS.	266
Integracja JMS i adaptory zasobów	267
Powiązania importu i eksportu JMS	267
Nagłówki JMS	269
Schemat korelacji tymczasowego, dynamicznego miejsca docelowego odpowiedzi JMS	270
Klienci zewnętrzne	271
Rozwiązywanie problemów z powiązaniem JMS	272
Obsługa wyjątków	273
Ogólne powiązania JMS	273
Przegląd ogólnych powiązań JMS.	274
Kluczowe opcje ogólnych powiązań usług JMS	277
Ogólne nagłówki JMS	278
Rozwiązywanie problemów z ogólnymi powiązaniem JMS	279
Obsługa wyjątków	280
Powiązania WebSphere MQ JMS	281
Przegląd powiązań WebSphere MQ JMS.	281
Kluczowe opcje powiązań usług JMS z produktem WebSphere MQ.	283
Nagłówki JMS	285
Klienci zewnętrzne	286
Rozwiązywanie problemów z powiązaniem JMS produktu WebSphere MQ.	286
Obsługa wyjątków	287
Powiązania produktu WebSphere MQ	288
Przegląd powiązań WebSphere MQ	288
Kluczowe opcje powiązania z produktem WebSphere MQ	290
Nagłówki WebSphere MQ	292
Statyczne dodawanie nagłówka MQCIH w wiazaniu WebSphere MQ	294
Klienci zewnętrzne	294

Rozwiązywanie problemów z powiązaniem produktu WebSphere MQ	295	Magistrala usług korporacyjnych w produkcie IBM	
Obsługa wyjątków	296	Business Process Manager	309
Ograniczenia powiązań	297	Łączenie usług za pośrednictwem magistrali usług korporacyjnych	309
Ograniczenia powiązania MQ	297	Infrastruktura przesyłania komunikatów magistrali usług korporacyjnych	311
Ograniczenia powiązań JMS, MQ JMS i ogólnych powiązań JMS	297	Hosty przesyłania komunikatów lub miejsca docelowego kolejki	311
Obiekty biznesowe	298	Składnice danych.	311
Definiowanie obiektów biznesowych	299	Dostawcy JDBC	312
Praca z obiektami biznesowymi	300	Magistrala integracji usług dla produktu IBM	
Specjalne obiekty biznesowe	301	Business Process Manager	312
Tryb analizowania obiektu biznesowego	302	Aplikacje usług i moduły usług	312
Uwagi na temat wybierania trybu analizowania obiektu biznesowego	302	Importy i powiązania importu	313
Zalety stosowania opóźnionego (zamiast przyspieszonego) trybu analizowania	303	Eksporty i powiązania eksportów	315
Uwagi na temat migrowania i programowania aplikacji	304	Moduły mediacji	316
Relacje.	305	Operacje podstawowe mediacji	318
Usługa relacji	308	Dynamiczne kierowanie.	323
Menedżer relacji	308	Sterowanie żądaniami usług przy użyciu strategii mediacji	324
Relacje w środowiskach wdrożenia sieciowego	309	WebSphere Service Registry and Repository	324
Funkcje API usługi relacji	309	WebSphere eXtreme Scale	325
		Klienci usługi komunikatów	325

Rozdział 1. Pierwsze kroki w programie IBM Business Process Manager

Bardzo ważne jest zrozumienie, jakie możliwości zarządzania procesami biznesowymi udostępnia produkt IBM® Business Process Manager oraz jakie są zależności między różnymi fazami zarządzania procesami biznesowymi, takimi jak tworzenie i wdrażanie aplikacji procesów.

Aplikacja procesów jest podstawowym kontenerem procesów i ich komponentów w produkcie IBM Business Process Manager. Programiści procesów tworzą aplikacje procesów w środowiskach tworzenia treści. Mogą one zawierać usługi, zadania i artefakty niezbędne podczas wykonywania.

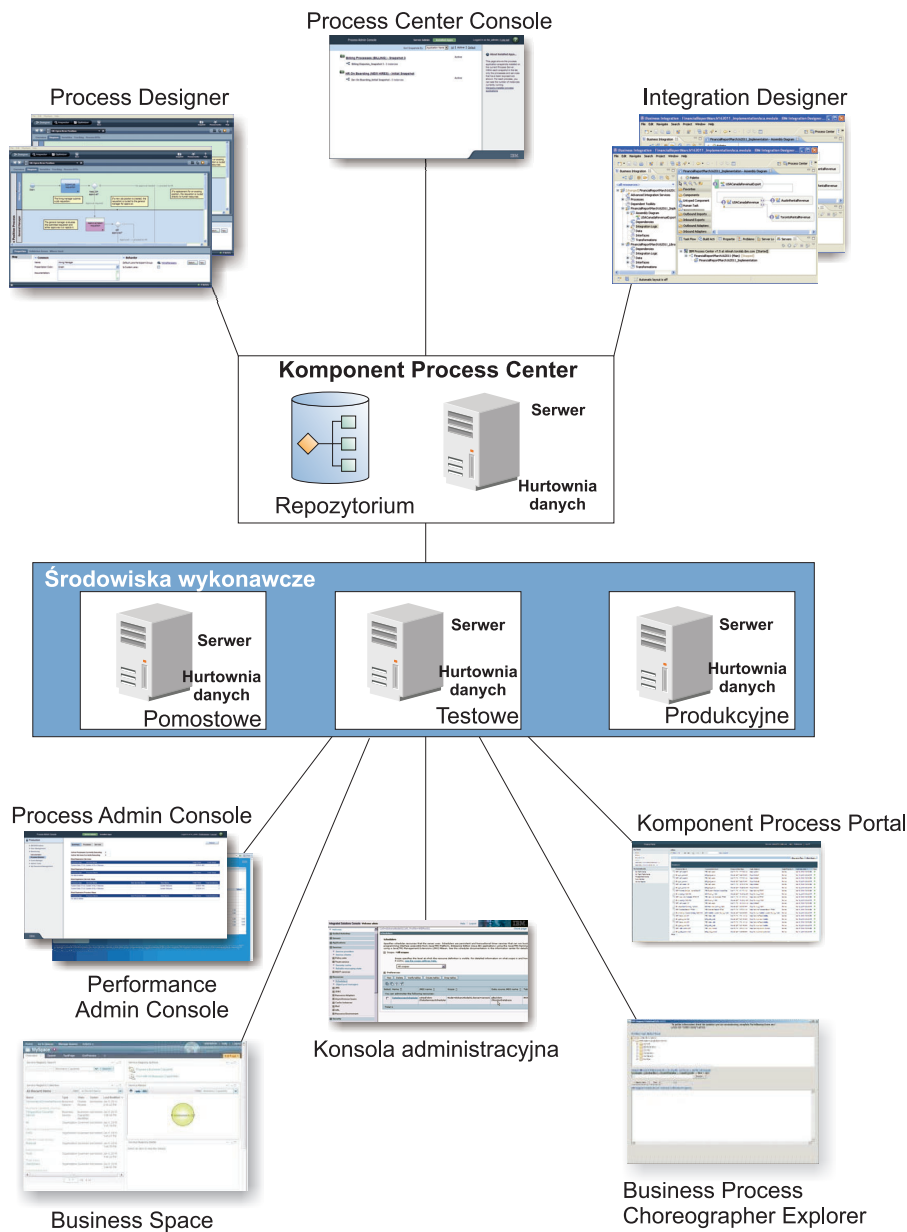
Zaawansowane usługi integracji zostały zaimplementowane w produkcie IBM Integration Designer i powiązane z aplikacjami procesów. Aplikacje procesów są wdrażane z komponentu Process Center do komponentu Process Server, który jest środowiskiem wykonawczym procesów dla produktu IBM Business Process Manager.

Podobnie zautomatyzowane procesy utworzone w komponencie Integration Designer mogą używać usług personelu utworzonych w produkcie IBM Process Designer.

Przegląd produktu

Produkt IBM Business Process Manager to wszechstronna platforma do zarządzania procesami biznesowymi umożliwiająca obserwację procesów biznesowych i zarządzanie nimi. Udostępnia on narzędzia i środowisko wykonawcze na potrzeby projektowania, wykonywania, monitorowania i optymalizowania procesów, a także podstawową obsługę integracji systemów. Produkt można skonfigurować tak, aby obsługiwał różne poziomy złożoności i podstawowe obszary zastosowań w zarządzaniu procesami biznesowymi.

Komponenty produktu IBM Business Process Manager udostępniają ujednoczone repozytorium BPM, narzędzia dla autorów, administratorów i użytkowników oraz platformę wykonawczą. Produkt można skonfigurować tak, aby obsługiwał różne poziomy złożoności i podstawowe obszary zastosowań w zarządzaniu procesami biznesowymi. Poniższy diagram ilustruje typową konfigurację produktu IBM Business Process Manager:



- W środowiskach tworzenia treści w komponentach IBM Process Designer i IBM Integration Designer programiści nawiązują połączenie z komponentem IBM Process Center. Z poziomu dowolnego z tych narzędzi programistycznych wyposażonych w graficzny interfejs użytkownika programiści mogą tworzyć, testować, debugować i wdrażać aplikacje biznesowe. Narzędzie można wybrać w zależności od typu tworzonej aplikacji. Mogą również wystąpić przypadki, w których najbardziej korzystne jest stosowanie obu tych narzędzi.
- W środowiskach tworzenia treści w komponentach Process Designer i Integration Designer projektanci procesów i usług tworzą możliwe do wdrożenia aplikacje procesów i biblioteki narzędziowe wielokrotnego użytku. Aplikacje procesów zawierają modele procesów i implementacje usług, w tym wymagane pliki dodatkowe. Aplikacje procesów są przechowywane w repozytorium komponentu Process Center, co umożliwia ich współużytkowanie.
- Komponent Process Center zawiera dwa serwery: serwer Process Center i serwer Performance Data Warehouse. Te serwery pozwalają programistom pracującym w komponente Process Designer na uruchamianie własnych aplikacji procesów oraz zapisywanie danych wydajności na potrzeby testowania i odtwarzania w trakcie prac programistycznych. Komponent Data Warehouse pobiera śledzone dane w regularnych odstępach czasu z serwera Process Server lub Process Center.
- Komponent Process Center obsługuje również wiele funkcji administracyjnych. Konsola Process Center Console umożliwia administratorom instalowanie na serwerach Process Server aplikacji procesów, które są gotowe do

wdrożenia w środowiskach pomostowych, testowych i produkcyjnych. Administratorzy mogą również zarządzać działającymi instancjami aplikacji procesów w skonfigurowanych środowiskach.

- Aplikacje procesów są instalowane na serwerze Process Server w środowiskach pomostowych, testowych i produkcyjnych. Środowiska wykonawcze obsługują procesy BPMN (Business Process Model and Notation) 2.0. Produkt IBM Business Process Manager Advanced obsługuje również procesy BPEL (Business Process Execution Language).
- Konsole Process Admin Console i Performance Admin Console umożliwiają administratorom zarządzanie wszystkimi serwerami wykonawczymi i ich obsługę. Konsoli Process Admin Console można użyć do zarządzania serwerem Process Center i serwerami Process Server w środowiskach wykonawczych. Konsoli Performance Admin Console można użyć do identyfikowania wąskich gardeł wydajności oraz przechwytywania danych instrumentacji na potrzeby dalszej analizy.
- Konsoli administracyjnej można użyć do tworzenia obiektów takich jak zasoby, aplikacje i serwery, a także zarządzania nimi. Dodatkowo z Konsoli administracyjnej można korzystać do wyświetlania komunikatów produktu.
- Business Space służy do tworzenia niestandardowych obszarów biznesowych udostępniających widgety do monitorowania różnych aspektów systemu lub administrowania nimi. Można na przykład monitorować działania biznesowe, usługi i poprawność systemu lub administrować strategiami mediacji i kalendarzami biznesowymi. Istnieje również możliwość utworzenia obszaru biznesowego z widżetami Zarządzanie czynnościami personelu i użycia go do uczestniczenia w procesach biznesowych.
- Dzięki portalowi Process Portal uczestnicy procesu mogą nawiązywać połączenia z serwerem Process Center lub z serwerami Process Server w każdym skonfigurowanym środowisku wykonawczym w zależności od tego, czy proces jest opracowywany, testowany, czy opublikowany w środowisku produkcyjnym.
- Instancjami procesów BPEL (Business Process Execution Language) można zarządzać w eksploratorze Business Process Choreographer Explorer lub w produkcie Business Space.

Konfiguracje produktu IBM Business Process Manager

Różne konfiguracje produktu IBM Business Process Manager są skorelowane z typowymi punktami wejścia lub etapami rozwoju w firmowym programie zarządzania procesami biznesowymi.

Tabela 1. Konfiguracje produktu IBM Business Process Manager

Konfiguracja	Faza
Zaawansowana	<p>Transformacja</p> <p>Pelen zestaw możliwości do zarządzania procesami biznesowymi.</p> <ul style="list-style-type: none"> • Rozszerzona obsługa automatyki dużego wolumenu procesów. • Wbudowane komponenty SOA do rozszerzonej ogólnokorporacyjnej integracji usług, orkiestracja.
Standardowa	<p>Program</p> <p>Konfiguracja przeznaczona dla typowych projektów zarządzania procesami biznesowymi.</p> <ul style="list-style-type: none"> • Programy poprawy jakości i wydajności obejmujące wiele projektów, o dużym znaczeniu dla biznesu. • Podstawowa obsługa integracji systemów. • Szybkie osiągnięcie korzyści i poprawa produktywności użytkowników.
Ekspresowa	<p>Projekt</p> <p>Konfiguracja przeznaczona dla pierwszego projektu zarządzania procesami biznesowymi.</p> <ul style="list-style-type: none"> • Szybkie osiągnięcie korzyści i poprawa produktywności użytkowników. • Niska cena początkowa. • Łatwe instalowanie i konfigurowanie.

Możliwości konfiguracji produktu IBM Business Process Manager

Bardzo ważne jest zrozumienie, jakie produkty i możliwości oferuje IBM w zakresie zarządzania procesami biznesowymi, i wybranie rozwiązania najbardziej odpowiedniego dla przedsiębiorstwa.

IBM Business Process Manager jest pojedynczą platformą BPM, która łączy w sobie możliwości dotyczące ludzi i możliwości integracyjne. Aby zaspokoić potrzeby i wymagania różnych użytkowników, dostępne są różne konfiguracje produktu. Konfiguracje produktu można łączyć podczas grupowego tworzenia treści oraz w sieciowych środowiskach wykonawczych.

Tabela 2. Możliwości konfiguracji produktu IBM Business Process Manager

Możliwość	Adv.	Std.	Express
Wykonywanie zgodne z produktem WebSphere Lombardi Edition	X	X	X
Process Designer (BPMN)	X	X	X
Grupowe edytowanie / natychmiastowe odtwarzanie	X	X	X
Interfejsy użytkownika umożliwiające interaktywne „trenowanie procesu”	X	X	X
Reguły procesu oparte na logice ODM	X	X	X
Process Portal	X	X	X
Monitorowanie i raportowanie w czasie rzeczywistym	X	X	X
Analitik i optymalizator wydajności	X	X	X
Performance Data Warehouse	X	X	X
Process Center / repozytorium współużytkowanych zasobów aplikacyjnych	X	X	X
Nieograniczona liczba autorów i użytkowników końcowych	X	X	200 użytkowników / 3 autorów
Wysoka dostępność: technologia klastrowa i nieograniczona liczba rdzeni	X	X	<ul style="list-style-type: none"> • Środowisko produkcyjne obsługuje 4 rdzenie • Środowisko programistyczne obsługuje 2 rdzenie • Brak klastrów
Wykonywanie zgodne z produktem WebSphere Process Server	X		
Integration Designer (BPEL / SOA)	X		
Wbudowana magistrala ESB (enterprise service bus)	X		
Obsługa transakcji	X		
Adaptory integracji	X		
Elastyczny interfejs użytkownika produktu Business Space	X		
Zaawansowana obsługa platform (Linux w systemie System z, IBM AIX, Solaris)	X	X	*Uwaga

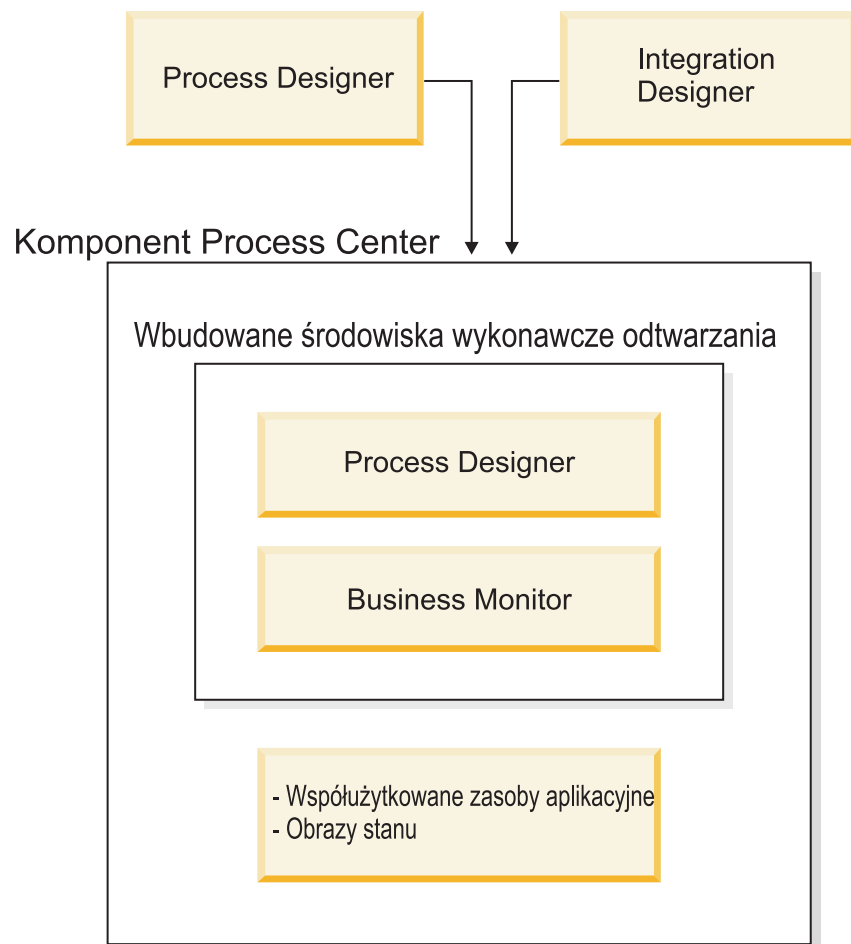
Uwaga: Produkt IBM BPM Express jest obsługiwany w systemie AIX tylko w przypadku klientów IBM Master Data Management (MDM).

Repozytorium komponentu Process Center

Process Center zawiera repozytorium dla wszystkich procesów, usług i innych zasobów aplikacyjnych utworzonych w środowiskach tworzenia treści programu IBM Business Process Manager oraz w produktach Process Designer i Integration Designer.

Komponent Process Center jest komponentem oprogramowania działającym jako serwer, w którym produkty Process Designer i Integration Designer współużytkują zasoby aplikacyjne, co umożliwia tym produktom wspólne opracowywanie procesów biznesowych w bardzo interaktywny sposób.

Na poniższym diagramie przedstawiono kilka powiązanych komponentów, które razem umożliwiają budowanie złożonych procesów biznesowych.



Konsola Process Center Console udostępnia narzędzia potrzebne do obsługi repozytorium.

- W konsoli Process Center Console można tworzyć aplikacje procesów i biblioteki narzędziowe oraz przydzielać do nich dostęp innym użytkownikom.
- W środowiskach tworzenia treści można tworzyć modele procesów, usługi i inne zasoby aplikacyjne wchodzące w skład aplikacji procesu.
- Komponent Process Center zawiera dwa serwery: serwer Process Center i serwer Performance Data Warehouse. Serwer Process Center pozwala programistom pracującym w produkcie Process Designer na uruchamianie własnych aplikacji procesów oraz zapisywanie danych wydajności na potrzeby testowania i odtwarzania w trakcie prac programistycznych. Komponent Performance Data Warehouse pobiera śledzone dane w regularnych odstępach czasu z serwera Process Server lub Process Center.
- Z poziomu konsoli Process Center administratorzy instalują aplikacje procesów gotowe do testowania lub produkcji na serwerach procesów w tych środowiskach.
- Z poziomu konsoli Process Center Console administratorzy zarządzają instancjami aplikacji procesu działającymi w skonfigurowanych środowiskach.

Konsola Process Center Console jest wygodnym miejscem, w którym można tworzyć i obsługiwać kontenery wysokiego poziomu, takie jak aplikacje procesów i biblioteki narzędziowe. Administratorzy, którzy nie korzystają aktywnie z widoku komponentu Designer, mogą za pomocą konsoli Process Center Console udostępnić środowisko umożliwiające analitykom BPM i programistom budowanie procesów i implementacji bazowych. Innym podstawowym zadaniem administratorów jest zarządzanie dostępem do repozytorium konsoli Process Center przez konfigurowanie odpowiedniej autoryzacji użytkowników i grup.

Użytkownicy z odpowiednimi uprawnieniami mogą wykonywać niektóre czynności administracyjne bezpośrednio w produktach Process Designer i Integration Designer. Na przykład programista z prawem do zapisu w aplikacji procesu, który chce przechwycić stan wszystkich zasobów aplikacyjnych projektu na znaczącym etapie projektowania, może utworzyć obraz stanu, pracując w widoku komponentu Designer.

Komponent Process Server a środowiska wykonawcze

Komponent Process Server udostępnia pojedyncze środowisko wykonawcze BPM, które może obsługiwać wiele procesów biznesowych, orkiestrację usług i możliwości integracji.

W środowiskach tworzenia treści zintegrowany serwer Process Server w komponencie Process Center umożliwia uruchamianie procesów podczas ich budowania. Można je uruchomić na serwerach Process Server w środowiskach wykonawczych. Komponent Business Performance Data Warehouse umożliwia gromadzenie i agregowanie danych dla procesów działających na serwerach Process Server. Zebrane dane można wykorzystać do ulepszenia procesów biznesowych.

Konsola Process Admin Console umożliwia zarządzanie serwerami Process Server w środowiskach wykonawczych, takich jak środowisko testowe, pomostowe czy produkcyjne, a serwer Process Server jest częścią komponentu Process Center.

Środowiska tworzenia treści

IBM Business Process Manager Advanced oferuje dwa środowiska tworzenia treści. Komponent IBM Process Designer służy do efektywnego modelowania procesów biznesowych, które wiążą się z czynnościami personelu. Komponent IBM Integration Designer służy do budowania usług, które są niezależne lub które wywołują inne istniejące usługi, takie jak usługi Web Services, aplikacje zasobów przedsiębiorstwa czy aplikacje działające w systemach CICS i IMS.

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

Process Designer

Process Designer jest dostępny we wszystkich edycjach produktu. IBM Business Process Manager Advanced zawiera również komponent Integration Designer oraz powiązane z nim edytory i adaptery.

Proces jest główną jednostką logiki w produkcie IBM Business Process Manager. Jest on kontenerem na wszystkie komponenty definicji procesu, w tym usługi, działania i bramki, zdarzenia licznika czasu, komunikatów i wyjątków, linie sekwencji, reguły oraz zmienne. Modelowanie procesu polega na tworzeniu definicji procesów biznesowych (business process definition - BPD) wielokrotnego użytku. Zarówno komponent Process Designer, jak i produkt Integration Designer umożliwiają tworzenie modeli procesów, które mogą zawierać czynności personelu.

Process Designer ułatwia tworzenie procesów biznesowych. Korzystając z łatwego w użyciu narzędzia graficznego, można tworzyć sekwencje działań, które składają się na proces biznesowy, a także modyfikować go, gdy po upływie pewnego czasu zmieniają się okoliczności. Jeśli jedno lub więcej działań wymaga dostępu do wielkich systemów zaplecza lub usług dostarczających dane do procesu biznesowego, na przykład aby pobrać informacje o klientach, być może trzeba będzie użyć komponentu Integration Designer. Korzystając z prostego interfejsu, w działaniu w komponencie Process Designer można wywołać usługę utworzoną w komponencie Integration Designer. Usługa ta może z kolei korzystać z przepływów mediacji do transformowania, kierowania i rozszerzania danych i adapterów i w ten sposób uzyskiwać dostęp do wielu systemów zaplecza w standardowy sposób. W skrócie, komponent Process

Designer skupia się na procesie biznesowym, a komponent Integration Designer — na zautomatyzowanych usługach uzupełniających proces biznesowy. Patrz temat Getting started with IBM Process Designer (Pierwsze kroki z produktem IBM Process Designer).

Wszystkie projekty komponentu Process Designer są zawarte w aplikacjach procesów. Aplikacje procesów i powiązane z nimi artefakty są przechowywane w repozytorium Process Center. Aplikacje procesów mogą współużytkować zasoby umieszczone w bibliotekach narzędziowych.

IBM Business Process Manager udostępnia szereg interfejsów użytkownika, które umożliwiają modelowanie, implementowanie, przeprowadzanie symulacji i badanie procesów biznesowych. W konsoli Process Center Console można tworzyć aplikacje procesów, biblioteki narzędziowe, ścieżki i obrazy stanu oraz zarządzać nimi. W komponencie Process Designer można tworzyć modele procesów, raporty i proste usługi. W komponencie Inspector można uruchamiać i debugować procesy. W komponencie Optimizer można przeprowadzać symulacje.

Aplikacje procesów utworzone w programie Process Designer można uruchamiać w dowolnym czasie na serwerze Process Center; można je również zachować w obrazie stanu i wdrażać na serwerze Process Server. To samo dotyczy usług utworzonych w komponencie Integration Designer i powiązanych z nimi aplikacji procesów.

Punkt rozszerzenia elementu uwierzytelniającego logowania został zdefiniowany w produkcie Process Designer w celu otwarcia punktu dostosowywania dla logiki logowania po stronie klienta na potrzeby spełnienia specjalnych wymagań dotyczących uwierzytelniania po stronie serwera. Po uruchomieniu uwierzytelniania produkt Process Designer pobiera element uwierzytelniający i wywołuje logikę logowania. Punkt rozszerzenia elementu uwierzytelniającego jest udostępniany w formacie wtyczki środowiska Eclipse. Aby korzystać z wtyczki elementu uwierzytelniającego, należy zapoznać się z sekcją Instalowanie produktu IBM Process Designer.

Integration Designer

Integration Designer udostępnia edytory i pomoce ułatwiające programistom tworzenie zautomatyzowanych złożonych procesów i usług (np. moduły SCA, mediacje i procesy BPEL). Jest on dostępny jako część pakietu IBM Business Process Manager Advanced lub jako samodzielny zestaw narzędzi do innych zastosowań.

IBM Integration Designer został zaprojektowany jako kompletne środowisko programowania integracji do budowania zintegrowanych aplikacji. Zintegrowane aplikacje nie są proste. Mogą wywoływać aplikacje w systemach EIS (Enterprise Information System), przeprowadzać procesy biznesowe obejmujące wiele wydziałów lub przedsiębiorstw, wywoływać — lokalnie lub zdalnie — aplikacje napisane w różnych językach i działające w różnych systemach operacyjnych. Komponenty można tworzyć i składać w inne zintegrowane aplikacje (czyli aplikacje utworzone z zestawu komponentów) za pomocą edytorów wizualnych. Edytory wizualne stanowią warstwę abstrakcji między komponentami i ich implementacjami. Programista używający tych narzędzi może złożyć zintegrowaną aplikację bez szczegółowej wiedzy na temat implementacji bazowych wszystkich komponentów.

Narzędzia produktu Integration Designer mają architekturę zorientowaną na usługi. Komponenty są usługami, zintegrowana aplikacja zawierająca wiele komponentów również jest usługą. Utworzone usługi są zgodne z czołowymi standardami branżowymi. Procesy BPEL, które również stają się komponentami, są tworzone w podobny sposób, przy użyciu prostych narzędzi wizualnych zgodnych z branżowym standardem BPEL (Business Process Execution Language).

Zgodnie z paradygmatem produktu Integration Designer, komponenty są składane w moduły. Importy i eksporty umożliwiają współużytkowanie danych przez różne moduły. Artefakty umieszczone w bibliotece również mogą być współużytkowane między modułami.

Moduły i biblioteki mogą być powiązane z aplikacją procesu i używane na serwerze Process Center oraz mogą być używane jako usługi przez procesy utworzone w komponencie Process Designer. W takich przypadkach można je wdrażać razem z aplikacją procesu.

Moduły i biblioteki można także wdrażać bezpośrednio w środowisku testowym lub na serwerze Process Server. Modułów mediacji można używać do tworzenia przepływów mediacji, które następnie można wdrożyć w produkcie WebSphere Enterprise Service Bus lub na serwerze Process Server.

IBM Integration Designer udostępnia także możliwość tworzenia typów danych i odwzorowań XML, które można wdrożyć na urządzeniu WebSphere DataPower. Można również przysyłać pliki na urządzenie WebSphere DataPower i z niego.

Narzędzia administracyjne

IBM Business Process Manager obejmuje zestaw narzędzi administracyjnych ułatwiających wykonywanie takich czynności, jak instalowanie obrazów stanu i zarządzanie nimi czy administrowanie procesami i praca z zasobami w środowisku informatycznym.

Narzędzia wiersza komend

IBM Business Process Manager udostępnia narzędzia wiersza komend, interfejsy skryptowe oraz interfejsy programistyczne służące do administrowania środowiskiem wykonawczym.

- Narzędzia wiersza komend są prostymi programami uruchamianymi w wierszu komend systemu operacyjnego w celu wykonania określonych czynności. Narzędzia te umożliwiają uruchamianie i zatrzymywanie serwerów aplikacji, sprawdzanie statusu serwera, dodawanie lub usuwanie węzłów oraz wykonywanie innych czynności.
- Administracyjny program skryptowy produktu WebSphere (wsadmin) jest znakowym środowiskiem interpretera komend pozwalającym na uruchamianie opcji administracyjnych w języku skryptowym (Jython lub Jacl) oraz na wykonywanie programów napisanych w języku skryptowym. Obsługuje on te same czynności, co Konsola administracyjna, jak również wiele spośród zadań wykonywanych przez konsolę Process Center Console. Narzędzie wsadmin jest przeznaczone do użycia w środowiskach produkcyjnych w celu wykonywania operacji nienadzorowanych.
- Administracyjne interfejsy programistyczne są zestawem klas i metod języka Java zgodnych ze specyfikacją Java Management Extensions (JMX), które zapewniają obsługę administrowania architekturą Service Component Architecture (SCA) oraz obiektami biznesowymi. Każdy interfejs programistyczny jest wyposażony w opis jego przeznaczenia, przykład demonstrujący użycie interfejsu lub klasy oraz odniesienia do opisów poszczególnych metod.

Process Center Console

Konsola Process Center Console jest wygodnym miejscem, w którym użytkownicy mogą tworzyć i obsługiwać biblioteczne elementy wysokiego poziomu, takie jak aplikacje procesów i biblioteki narzędziowe. Ułatwia zapewnienie środowiska, w którym analitycy i programiści produktu BPM mogą budować procesy i implementacje bazowe. Oprócz tego konsola Process Center Console udostępnia narzędzia służące do obsługi repozytorium, w tym konfigurowania odpowiedniej autoryzacji dla użytkowników i grup.

Dostęp do konsoli Process Center Console można uzyskać za pomocą przeglądarki WWW (na przykład <http://host:9080/ProcessCenter>).

Process Admin Console

Konsola Process Admin Console służy do administrowania serwerami procesów w środowisku, w tym użytkownikami i zainstalowanymi obrazami stanu na każdym serwerze. Oprócz tego udostępnia narzędzia ułatwiające zarządzanie kolejkami i pamięciami podręcznymi.

Konsola Process Admin Console obejmuje narzędzie Process Inspector, za pomocą którego użytkownicy mogą wyświetlać instancje procesów związane z aplikacjami procesów działającymi na określonym serwerze procesów oraz nimi zarządzać.

Dostęp do konsoli Process Admin Console można uzyskać za pomocą przeglądarki WWW (na przykład <http://host:9080/ProcessAdmin>).

Business Performance Admin Console

Konsola Business Performance Admin Console obejmuje narzędzia umożliwiające zarządzanie komponentami Performance Data Warehouse w środowisku użytkownika. Za pomocą tego narzędzia można zarządzać kolejkami serwera oraz monitorować wydajność serwera.

Dostęp do konsoli Business Performance Admin Console można uzyskać za pomocą przeglądarki WWW (na przykład <http://host:9080/PerformanceAdmin>).

Konsola administracyjna serwera WebSphere Application Server

Konsola administracyjna jest używana do administrowania aplikacjami, usługami i innymi zasobami w zasięgu komórki, węzła, serwera lub klastra. Konsoli tej można używać z serwerami autonomicznymi oraz menedżerami wdrażania zarządzającymi wszystkimi serwerami w komórce środowiska sieciowego.

Zainstalowanie profilu autonomicznego oznacza, że istnieje pojedynczy węzeł w swojej własnej domenie administracyjnej nazywanej komórką. Konsola administracyjna umożliwia zarządzanie aplikacjami, magistralami, serwerami i zasobami w obrębie tej domeny administracyjnej. Podobnie zainstalowanie i skonfigurowanie komórki wdrożenia sieciowego oznacza, że w tej samej komórce istnieje węzeł menedżera wdrażania oraz przynajmniej jeden węzeł zarządzany. Konsola administracyjna umożliwia zarządzanie aplikacjami, konfigurowanie węzłów zarządzanych w komórce, a także monitorowanie tych węzłów i ich zasobów oraz sterowanie nimi.

Dostęp do tej konsoli można uzyskać za pomocą przeglądarki WWW (na przykład <http://host:9060/ibm/console> lub <https://host:9043/ibm/console>).

Business Process Choreographer Explorer i Business Process Archive Explorer

W zależności od pełnionej roli użytkownik może używać tych interfejsów klienckich do zarządzania procesami BPEL i czynnościami personelu utworzonymi w produkcie IBM Integration Designer, pracy z przypisanymi czynnościami, wyświetlania zakończonych procesów BPEL i czynności personelu, które znajdują się w archiwalnej bazie danych, lub usuwania procesów i czynności z archiwum.

Widżety administracyjne

Widżety administracyjne umożliwiają monitorowanie niektórych komponentów rozwiązania do zarządzania procesami biznesowymi, w tym modułów i usług zaawansowanej usługi integracji, oraz zarządzanie tymi komponentami. Tych widżetów można użyć w obszarze biznesowym, aby zapewnić widoczność aplikacji usługi i modułów, a także aby odpowiedzieć na pytania takie jak podane poniżej:

- Jakie usługi są konsumowane lub ujawniane w module oraz jakie wartości mają czas odpowiedzi i przepustowość dla tych usług w zdefiniowanym okresie?
- Jaki jest status modułu?
- Czy w module istnieją jakieś zdarzenia zakończone niepowodzeniem?
- Jakie strategie mediacji zostały powiązane z modulem?
- Jakie procesy BPEL i czynności personelu są używane w module?
- Czy w module istnieją jakiegokolwiek kalendarze biznesowe lub reguły biznesowe?

Aby uzyskać obraz stanu ogólnej poprawności systemu dla rozwiązania biznesowego, w tym informacje na temat statusu topologii (środowisk wdrażania, klastrów), aplikacji systemowych (na przykład menedżera zdarzeń zakończonych niepowodzeniem lub produktu Business Process Choreographer), źródeł danych, mechanizmów przesyłania komunikatów i kolejek przesyłania komunikatów, należy użyć co najmniej jednego z tych widżetów.

Menedżer reguł procesów biznesowych

Menedżer reguł procesów biznesowych jest narzędziem z interfejsem WWW ułatwiającym analitykom biznesowym przeglądanie i modyfikowanie wartości reguł biznesowych. To narzędzie jest opcją komponentu IBM Process Server, której instalację można wybrać podczas tworzenia profilu lub po zainstalowaniu serwera.

Ułatwienia dostępu w produkcie IBM Business Process Manager

Ułatwienia dostępu są to funkcje, których zadaniem jest pomoc osobom niepełnosprawnym, na przykład z ograniczonymi możliwościami ruchowymi lub z poważnymi wadami wzroku, w efektywnym korzystaniu z produktów technologii informatycznych.

IBM stara się udostępniać produkty, których może używać większość osób, niezależnie od wieku i możliwości fizycznych.

Szczegółowe informacje na temat ułatwień dostępu w tym produkcie zawiera sekcja Ułatwienia dostępu w produkcie IBM Business Process Manager.

Dostępność języków narodowych w produkcie IBM Business Process Manager

Produkt IBM Business Process Manager jest dostępny w różnych językach. Lista zawiera informacje o poziomie obsługi dla poszczególnych języków.

IBM Business Process Manager obsługuje następujące języki. Dokumentacja może nie być przetłumaczona w pełni.

- chiński uproszczony
- chiński tradycyjny
- czeski
- angielski (Stany Zjednoczone)
- francuski
- niemiecki
- węgierski
- włoski
- japoński
- koreański
- polski
- portugalski (Brazylia)
- rosyjski
- hiszpański

IBM Business Process Manager zapewnia częściową obsługę następujących języków. Dokumentacja może nie być przetłumaczona w pełni.

- arabski (przetłumaczono widgety czynności personelu BPEL, widgety eksploratora produktu Business Process Choreographer, środowisko Business Space oraz widgety monitora Business Space)
- duński (przetłumaczono widgety monitora Business Space oraz środowisko Business Space)
- holenderski (przetłumaczono produkt Process Designer, komponent Process Center, projektant definicji BPD, wtyczkę JSEditor, środowisko Business Space oraz widgety monitora Business Space)
- fiński (przetłumaczono widgety monitora Business Space, środowisko Business Space, projektant definicji BPD, projektant usług, wtyczkę JSEditor i produkt Process Designer)
- grecki (przetłumaczono komponenty Process Designer, Process Center oraz Business Space)
- hebrajski (przetłumaczono czynności personelu BPEL, eksplorator produktu Business Process Choreographer, środowisko Business Space oraz widgety monitora Business Space)
- norweski (przetłumaczono widgety monitora Business Space oraz środowisko Business Space)
- portugalski - Portugalia (produkt Process Designer, komponent Process Center, projektant definicji BPD, projektant usług i wtyczka JSEditor)
- rumuński (przetłumaczono operacje środowiska wykonawczego)
- słowacki (przetłumaczono produkt Business Space, projektant definicji BPD, projektant usług, wtyczkę JSEditor i produkt Process Designer)
- szwedzki (przetłumaczono widgety monitora Business Space oraz środowisko Business Space)
- turecki (przetłumaczono produkt Business Space)

Uwaga: W przypadku tureckich ustawień narodowych konieczne jest ustawienie wpisu **case-insensitive-security-cache** w pliku **60Database.xml** na wartość **false**, aby nazwy użytkowników i hasła mogły zawierać literę *i*. Plik **60Database.xml** znajduje się w katalogu *instalacyjny_katalog_główny\profiles\nazwa_profilu\config\cells\nazwa_komórki\nodes\nazwa_węzła\servers\nazwa_serwera\process-center\config\system*.

IBM Business Process Manager umożliwia wprowadzanie przez użytkowników łańcuchów dwukierunkowych w środowisku komponentu Process Designer, w formatkach i w komponencie Process Portal. Udostępnia interfejs API języka JavaScript zapewniający testową obsługę manipulowania językami dwukierunkowymi.

W formacie oraz w komponencie Process Portal wprowadzono obsługę kalendarzy hebrajskiego i arabskiego.

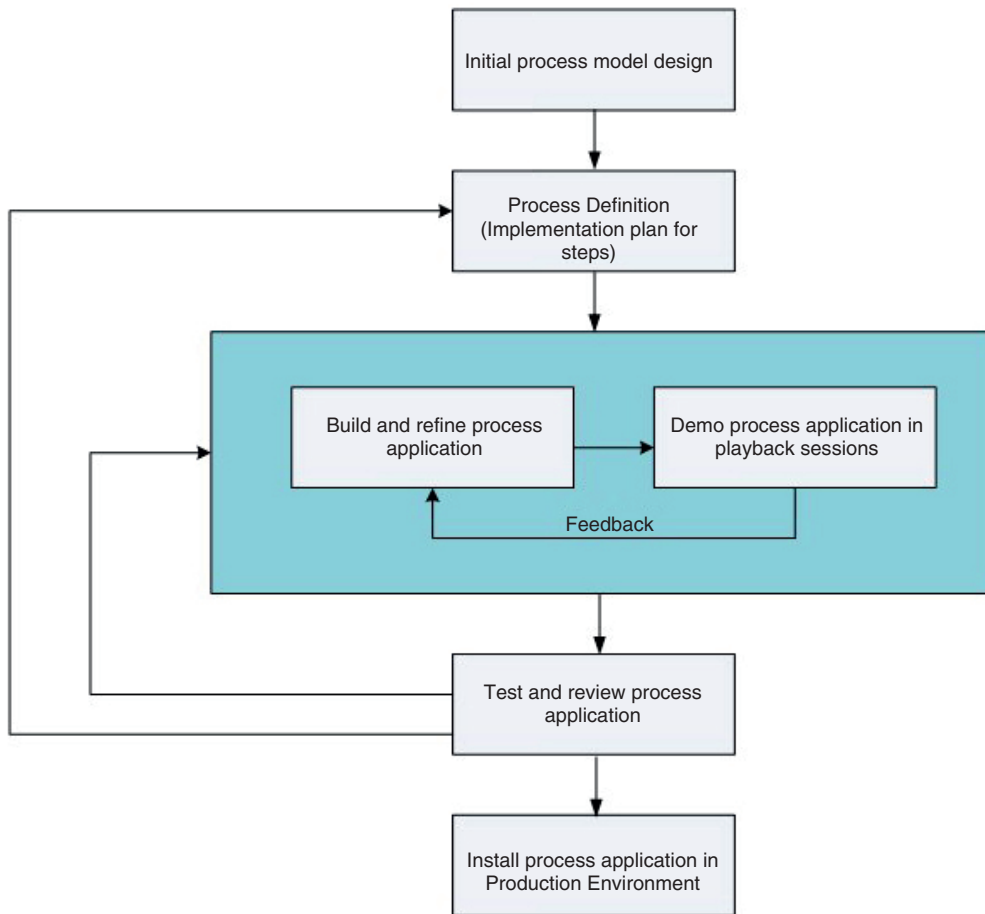
Przegląd zarządzania procesami biznesowymi

Programując procesy w komponencie Process Designer, należy zaplanować docelową instalację aplikacji procesów na serwerach w środowiskach testowych i produkcyjnych.

Komponent Process Designer znajduje się w produkcie IBM Business Process Manager Express, produkcie IBM Business Process Manager Standard i produkcie IBM Business Process Manager Advanced. W tej sekcji opisana zostanie wersja Advanced, która jest zaprojektowana pod kątem automatyzacji przy dużej objętości danych i korzystania z usług złożonych i utworzonych w produkcie Integration Designer. Ma ona wbudowane komponenty architektury SOA, których można używać na potrzeby ogólnokorporacyjnej integracji usług. Wersja Standard może być używana przez wielu specjalistów biznesowych jednocześnie do tworzenia wielu zaawansowanych procesów. Jest ona wyposażona w podstawowe mechanizmy integracji systemów. Wersja Express jest przeznaczona dla niewielkiej liczby użytkowników pracujących na pojedynczym serwerze, którzy rozpoczynają pracę z procesami biznesowymi lub nie wymagają dostępu do wielu systemów zewnętrznych.

Następujący diagram pokazuje cykl życia typowego programowania procesu. Uwzględniono tu również kroki związane z budowaniem i udoskonalaniem usługi instalacji, dzięki której aplikacje procesów można instalować w środowisku produkcyjnym.

Jak widać na diagramie, można pracować wyłącznie w środowisku programistycznym. Jednak serwery Process Server należy skonfigurować dla obu środowisk: testowego i produkcyjnego.



Przegląd modelowania procesów

Proces jest główną jednostką logiki w produkcie IBM Business Process Manager. Jest on kontenerem na wszystkie komponenty definicji procesu, w tym usługi, działania i bramki, zdarzenia licznika czasu, komunikatów i wyjątków, linie sekwencji, reguły oraz zmienne. Modelowanie procesu polega na tworzeniu definicji procesów biznesowych (Business Process Definition - BPD) wielokrotnego użytku.

Komponenty procesu pozwalają definiować przepływ pracy procesu dla użytkowników końcowych, tworzyć logikę w obrębie procesu oraz integrować z innymi aplikacjami i źródłami danych. Aby zrozumieć, co się dzieje w procesie w czasie wykonywania, należy dokładnie poznać komponenty, które się na niego składają w czasie projektowania.

Budowanie procesów w produkcie IBM BPM

W programowaniu procesu przy użyciu produktu IBM BPM zaangażowanych jest zwykle wiele osób z różnych organizacji. Ich główną troską jest zbudowanie najlepszego możliwego rozwiązania spełniającego określone cele projektu. Aby to osiągnąć, członkowie zespołu powinni współpracować przy określaniu wymagań procesu oraz iteracyjnym opracowywaniu modelu i jego implementacji.

Ponowne wykorzystanie elementów w produkcie Process Designer

Produkt Process Designer pozwala programistom procesów ponownie wykorzystać istniejące elementy zarówno w tej samej, jak i w innych aplikacjach procesów. Jeśli na przykład wiadomo, że istnieje już kilka usług obejmujących formatki i inne współużytkowane elementy potrzebne wielu użytkownikom, można zapewnić do nich dostęp i pozwolić na ich ponowne wykorzystywanie, umieszczając je w bibliotece narzędziowej. Umożliwi to dodanie do aplikacji

procesu zależności od biblioteki narzędziowej, w której znajdują się współużytkowane elementy. Dzięki temu użytkownik może wybrać jedną z istniejących usług, decydując się na konkretną implementację działania. Elementy znajdujące się w bibliotece narzędziowej mogą być także używane przez programistów pracujących nad innymi aplikacjami procesów.

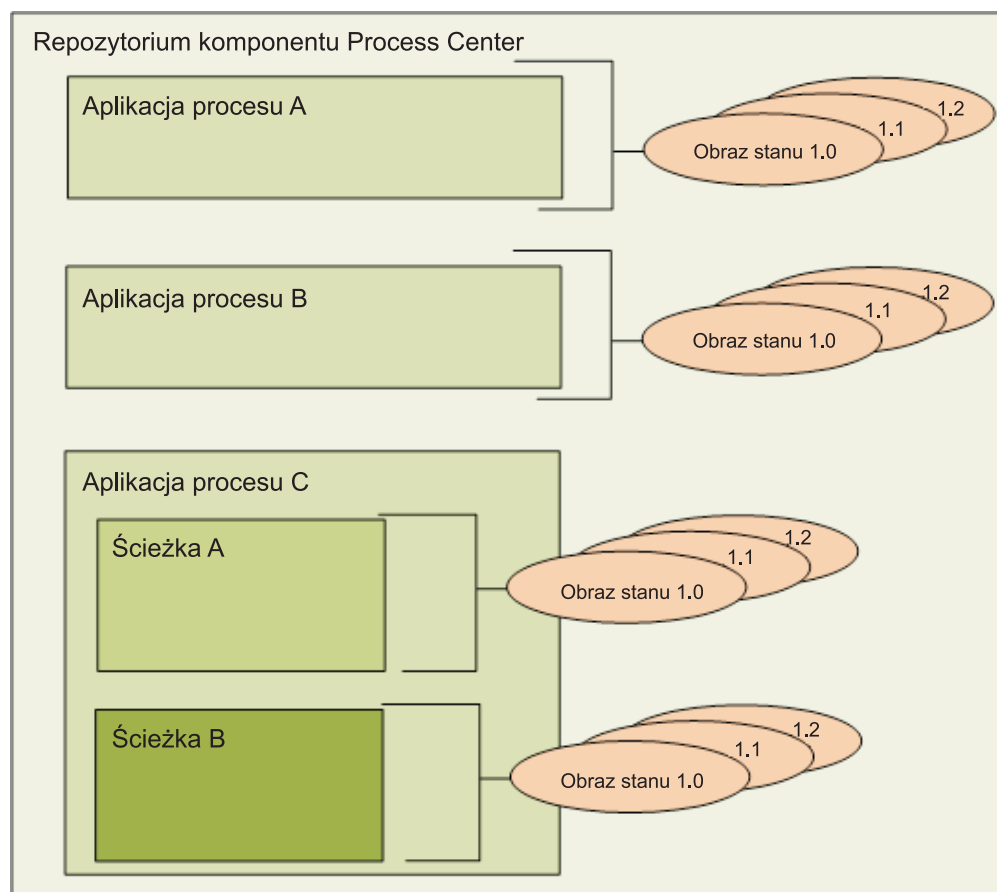
Korzystanie z komponentu Designer w produkcie IBM Process Designer

Interfejs komponentu Designer udostępnia narzędzia potrzebne przy modelowaniu procesów w produkcie IBM BPM.

Programowanie procesów z użyciem komponentu Process Center

Komponent IBM Process Center pełni rolę centralnego repozytorium dla wszystkich zasobów aplikacyjnych projektów tworzonych w produkcie Process Designer. Kiedy z komponentem Process Center łączy się wiele klientów produktu Process Designer, użytkownicy mogą współużytkować takie elementy, jak procesy i usługi. Są dla nich także widoczne zmiany wprowadzane przez innych użytkowników. Komponent Process Center może także służyć jako repozytorium zasobów aplikacyjnych utworzonych w produkcie IBM Integration Designer.

Podczas programowania procesów w produkcie Process Designer w repozytorium komponentu Process Center dostępna jest hierarchia zaprojektowana z myślą o ułatwieniu zarządzania projektami. Poniższa ilustracja przedstawia w ogólnym zarysie hierarchię repozytorium:



Z powyższego diagramu wynika, że repozytorium komponentu Process Center obejmuje następujące artefakty:

Typ zawartości	Opis
Aplikacje procesów	Kontenery modeli procesów i obsługujących je implementacji tworzonych przez analityków BPM i programistów w komponencie Designer produktu IBM Process Designer.

Typ zawartości	Opis
Ścieżki	Opcjonalne dodatkowe podziały w aplikacji procesu tworzone na podstawie czynności zespołu lub wersji aplikacji procesów. Kiedy ścieżki są włączone, umożliwiają równoległe prowadzenie prac. O tym, czy dodatkowe ścieżki są potrzebne i należy je włączyć dla danej aplikacji procesu, decydują administratorzy.
Obrazy stanu	Rejestrują stan elementów w aplikacji procesu lub ścieżce w określonym momencie. Obrazy stanu reprezentują zwykle kamienie milowe lub używane są na potrzeby odtwarzania bądź instalowania. Użytkownicy mogą porównywać i przywracać wcześniejsze obrazy stanu. Jeśli administrator włączy ścieżki dla aplikacji procesu, wtedy bazą dla nowej ścieżki jest obraz stanu.

Aplikacje procesów: przegląd

Aplikacja procesu jest kontenerem dla modeli procesów i obsługujących je implementacji. Jest ona przechowywana w repozytorium. Po utworzeniu artefaktów są one składane w aplikację procesu.

Aplikacje procesów mogą zawierać wszystkie poniższe artefakty lub tylko niektóre z nich:

- Co najmniej jeden model procesu. Modele procesów są zwane również definicjami procesów biznesowych (Business Process Definition - BPD).
- Odwołania do bibliotek narzędziowych.
- Usługi wymagane do zaimplementowania działań lub zintegrowania z innymi systemami, w tym zaawansowane usługi integracji.
- Przynajmniej jedna ścieżka.
- Moduły architektury SCA (Service Component Architecture) (opracowane w produkcie IBM Integration Designer).
- Dowolne inne elementy wymagane do uruchomienia procesu.

Film wideo wprowadzający do programowania iteracyjnego aplikacji procesów i bibliotek narzędziowych ze wskazówkami, obrazami stanu i ścieżkami znajduje się w sekcji “Iterative Process Application and Toolkit Development” dostępnej w serwisie YouTube lub Centrum informacyjnym produktu IBM Education Assistant. Dostępna jest transkrypcja tego filmu wideo.

Najnowsza wersja aplikacji procesu, obrazy stanu i ścieżki

Wszystkie zmiany dokonane w aplikacji procesu są dynamicznie zapisywane do repozytorium komponentu Process Center w najnowszej wersji, która jest bieżącą wersją roboczą aplikacji procesu. Sesje odtwarzania dla najnowszej wersji umożliwiają szybkie testowanie bieżącej wersji roboczej aplikacji procesu oraz zarządzanie nią.

Aplikacja procesu pozostaje na poziomie tej najnowszej wersji do czasu, aż użytkownik postanowi utworzyć obraz stanu, w którym rejestrowany jest stan elementów biblioteki w aplikacji procesu lub ścieżce w określonym momencie. Obraz stanu jest zazwyczaj tworzony za każdym razem, kiedy użytkownik jest gotowy do przetestowania integracji albo kiedy chce zainstalować aplikację procesu na serwerze Process Center lub serwerze Process Server na potrzeby środowiska testowego, pomostowego lub produkcyjnego.

Uwaga: Najnowsza wersja jest to specjalny obraz stanu. Jest to jedyny typ obrazu stanu, którego zawartość można zmieniać, ale który można uruchamiać tylko na serwerze Process Center. Najnowszej wersji nie można zainstalować na serwerze Process Server.

Każda aplikacja procesu ma domyślnie pojedynczą ścieżkę o nazwie Główna. Aby możliwe było równoległe tworzenie aplikacji procesu, można utworzyć dodatkowe ścieżki. Te opcjonalne podziały aplikacji procesu pozwalają izolować zmiany. Załóżmy na przykład, że w firmie trwa proces zmiany marki. W czasie tego przejścia muszą być zachowane bieżące aplikacje procesów, a jednocześnie tworzone są nowe wersje oparte na zaktualizowanej tożsamości korporacyjnej. W takiej sytuacji jeden zespół może wprowadzać niewielkie poprawki w bieżącej wersji aplikacji procesu (w ścieżce Główna), a inny zespół może tworzyć nową wersję aplikacji procesu w oddzielnej ścieżce.

Biblioteki narzędziowe dla aplikacji procesów

Biblioteki narzędziowe są kontenerami przechowującymi elementy biblioteki (na przykład definicje BPD) na potrzeby ponownego wykorzystania przez aplikacje procesów lub inne biblioteki narzędziowe. Aplikacje procesów mogą współużytkować elementy biblioteki z co najmniej jednej biblioteki narzędziowej, a biblioteki narzędziowe mogą współużytkować elementy biblioteki z innych bibliotek narzędziowych. Jeśli użytkownik ma dostęp do biblioteki narzędziowej, można utworzyć zależność od tej biblioteki narzędziowej i korzystać z jej elementów biblioteki w aplikacji procesu.

Aplikacje procesów i aplikacje poziomu biznesowego

Dla aplikacji procesu istnieje aplikacja poziomu biznesowego (Business Level Application - BLA), która działa jako kontener dla aplikacji procesu oraz jej zasobów aplikacyjnych (do zasobów aplikacyjnych należą na przykład moduły SCA, biblioteki narzędziowe i biblioteki). Każdy obraz stanu aplikacji procesu ma własną aplikację poziomu biznesowego. Wiele czynności administracyjnych dotyczących obrazu stanu (na przykład zatrzymywanie lub uruchamianie go na serwerze produkcyjnym) jest realizowanych na poziomie aplikacji BLA, co pozwala szybciej i łatwiej administrować obrazem stanu i wszystkimi jego zasobami aplikacyjnymi.

Iteracyjne programowanie aplikacji procesów i bibliotek narzędziowych

Tabela 3. Wprowadzenie

Scena	Audio	Działanie na ekranie
1	Witamy we wprowadzeniu do programowania iteracyjnego aplikacji procesów i bibliotek narzędziowych w produkcie IBM Business Process Manager.	Na ekranie powitalnym zostanie wyświetlony tytuł filmu wideo, <i>Iteracyjne programowanie aplikacji procesów i bibliotek narzędziowych</i> , oraz podtytuł, <i>Informacje na temat używania najnowszych wersji, obrazów stanu i ścieżek</i> . Copyright 2013, IBM Corporation.
2	W tym filmie wideo zostaną przedstawione informacje na temat aplikacji procesów i bibliotek narzędziowych, a także o sposobie korzystania z najnowszych wersji, obrazów stanu i ścieżek do zarządzania ich projektowaniem i wdrażaniem.	Zostanie wyświetlona wypunktowana lista tematów omówionych w filmie wideo.
3	Procesy ewoluują w czasie wraz ze zmianą potrzeb danej organizacji. Podczas przechodzenia aplikacji procesów i bibliotek narzędziowych przez cykl projektowania, testowania i produkcji można zarządzać wersjami i zasobami aplikacyjnymi procesów za pomocą produktu IBM Business Process Manager.	Zostanie wyświetlony diagram cyklu projektowania iteracyjnego. Strzałki przemieszczają się na diagramie w celu pokazania cyklicznej natury procesu projektowania.
4	W produkcie IBM Business Process Manager aplikacje procesów pełnią rolę kontenerów dla modeli procesów i implementacji pomocniczych tworzonych przez programistów w produkcie IBM Process Designer.	Zostanie wyświetlona karta Aplikacje procesów w konsoli Process Center Console. Cursor zostanie umieszczony nad nazwą aplikacji procesu, a następnie nad odsyłaczem do miejsca, w którym użytkownik może otworzyć aplikację w produkcie IBM Process Designer. Na ekranie zostanie wyświetlony tekst "Aplikacja procesu: instalowalny kontener modelu procesu".
5	Biblioteki narzędziowe to kontenery, które zawierają te same artefakty co aplikacje procesów. Jednak w przeciwieństwie do aplikacji procesów nie mogą być instalowane i uruchamiane na serwerze procesów. Biblioteki narzędziowe przechowują artefakty, które są wielokrotnie używane przez co najmniej jedną aplikację procesu. Biblioteki narzędziowe są instalowane pośrednio z każdą aplikacją procesu, która odwołuje się do nich.	Zostanie wyświetlona karta Biblioteki narzędziowe w konsoli Process Center Console. Cursor zostanie umieszczony nad nazwą biblioteki narzędziowej, a następnie nad odsyłaczem do miejsca, w którym użytkownik może otworzyć bibliotekę w produkcie IBM Process Designer. Na ekranie zostanie wyświetlony tekst "Biblioteka narzędziowa: kontener artefaktów wielokrotnego użytku".

Tabela 4. Najnowsze wersje, obrazy stanu i ścieżki

Scena	Audio	Działanie na ekranie
6	<p>Produkt IBM Business Process Manager obsługuje podejście iteracyjne w przypadku programowania aplikacji procesów i bibliotek narzędziowych obejmujące następujące elementy:</p> <ul style="list-style-type: none"> • Najnowsza wersja, czyli bieżąca wersja robocza. • Obrazy stanu, które rejestrują stan elementów biblioteki w określonym momencie. • Ścieżki, czyli opcjonalne gałęzie w aplikacji procesu lub bibliotece narzędziowej, których można używać do jednoczesnego projektowania wielu wersji. 	Zostanie wyświetlona wypunktowana lista funkcji zawartych w produkcie IBM BPM, które obsługują iteracyjne programowanie aplikacji procesów i bibliotek narzędziowych.
7	Podczas projektowania aplikacji procesu lub biblioteki narzędziowej w produkcie IBM Process Designer wszystkie wprowadzone zmiany są zapisywane w repozytorium komponentu Process Center. Ta bieżąca wersja robocza aplikacji procesu lub biblioteki narzędziowej jest nazywana najnowszą wersją.	Zostanie wyświetlony diagram procesu w produkcie IBM Process Designer. Zostanie otwarty komponent diagramu, a słowo <i>dane</i> zostanie zmienione na słowo <i>data</i> w formatce. Zmiany zostaną następnie zapisane. Na ekranie zostanie wyświetlony tekst "Najnowsza wersja: bieżąca wersja robocza".
8	Możliwe jest użycie sesji odtwarzania względem najnowszej wersji w celu natychmiastowego przetestowania bieżącej wersji roboczej aplikacji procesu lub biblioteki narzędziowej. Najnowszą wersję można uruchomić tylko na serwerze komponentu Process Center. Najnowszych wersji nie można instalować na serwerze procesów.	Zostanie otwarta górna strona diagramu procesu i zostanie uruchomiona sesja odtwarzania. Spowoduje to otwarcie interfejsu komponentu Inspector i wyświetlenie zmienionej formatki w oknie przeglądarki. Na ekranie zostanie wyświetlony tekst "Najnowsza wersja: odtwarzanie w celu natychmiastowego przetestowania".
9	Po zakończeniu modyfikowania aplikacji procesu lub biblioteki narzędziowej można utworzyć ich obraz stanu. Obraz stanu rejestruje stan wszystkich elementów aplikacji procesu lub biblioteki narzędziowej w określonym momencie. Po utworzeniu obrazu stanu procesy i elementy pokrewne są nadal dostępne do edycji w najnowszej wersji. W produkcie Process Designer dostępne obrazy stanu są wyświetlane w sekcji Historia podwersji. Chociaż można wyświetlić te obrazy stanu, nie można ich edytować.	Okno przeglądarki zostanie zamknięte i zostanie utworzony obraz stanu aplikacji procesu. Zostanie wyświetlony interfejs produktu Designer, a kursor zostanie umieszczony nad nazwą nowego obrazu stanu w sekcji Historia podwersji. Na ekranie zostanie wyświetlony tekst "Obraz stanu: rejestruje stan wszystkich elementów".
10	Możliwe jest również porównywanie obrazów stanu w celu uzyskania informacji, takich jak czas ich utworzenia oraz wszystkie procesy, które zostały do nich dodane.	Zostanie wyświetlony widok porównania obrazów stanu. Kursor zostanie umieszczony nad datą utworzenia obrazu stanu oraz nazwą elementu w procesie, który został zmieniony. Na ekranie zostanie wyświetlony tekst "Porównanie zmian między obrazami stanu".
11	Podczas tworzenia nowego obrazu stanu biblioteki narzędziowej aplikacji procesów, które odwołują się do niej, nadal będą używać starego obrazu stanu. Po otwarciu aplikacji procesu w produkcie IBM Process Designer w celu jego edycji zostanie wyświetlone ostrzeżenie powiadamiające o dostępności nowej wersji biblioteki narzędziowej.	Zostanie otwarta konsola Process Center Console. Zostanie wyświetlona strona Biblioteki narzędziowej, a następnie zostanie otwarta biblioteka narzędziowa. Zostanie utworzony nowy obraz stanu biblioteki narzędziowej. Zostanie otwarta strona Aplikacje procesów, a aplikacja procesu zostanie otwarta w produkcie Process Designer. Kursor zostanie umieszczony nad wyświetlanym ostrzeżeniem i jego pokrewnymi elementami menu. Na ekranie zostanie wyświetlony tekst "Aplikacje procesów obsługują odwołania do starego obrazu stanu biblioteki narzędziowej".

Tabela 4. Najnowsze wersje, obrazy stanu i ścieżki (kontynuacja)

Scena	Audio	Działanie na ekranie
12	Jeśli aplikacja procesu lub biblioteka narzędziowa są gotowe do wdrożenia, administrator może zainstalować jeden lub więcej ich obrazów stanu na serwerze procesów za pomocą konsoli Process Center Console. Zainstalowane aplikacje procesów mogą odwoływać się tylko do obrazów stanu biblioteki narzędziowej, a nie do bieżącej wersji roboczej biblioteki narzędziowej.	Zostanie otwarta konsola Process Center Console, a następnie aplikacja procesu. Cursor zostanie umieszczony nad odsyłaczem do obrazu stanu, który administrator może kliknąć w celu zainstalowania obrazu stanu. Na ekranie zostanie wyświetlony tekst “Instalowanie obrazów stanu aplikacji procesów na serwerze procesów”.
13	Administratorzy mogą również zarządzać obrazami stanu z poziomu konsoli Process Center Console, tworząc, archiwizując i eksportując obrazy stanu.	Zostanie otwarte menu rozwijane obok nazwy obrazu stanu, a kursor zostanie umieszczony nad każdą opcją w menu. Na ekranie zostanie wyświetlony tekst “Zarządzanie obrazami stanu za pomocą konsoli Process Center Console”.
14	Domyślnie każda aplikacja procesu lub biblioteka narzędziowa zawierają pojedynczą ścieżkę o nazwie Główna. Administratorzy mogą zezwolić na tworzenie dodatkowych ścieżek, aby umożliwić programowanie równoległe. Te opcjonalne podziały izolują zmiany.	Zostanie otwarta strona zarządzania aplikacją procesu. Zostanie zaznaczona opcja “Zezwól użytkownikom na tworzenie ścieżek w tej aplikacji procesu”. Na ekranie zostanie wyświetlony tekst “Ścieżki: opcjonalne podziały izolujące zmiany”.
15	Nową ścieżkę można utworzyć na podstawie obrazu stanu. Powoduje to skopiowanie wybranego obrazu stanu do nowej ścieżki. Inne obrazy stanu ze ścieżki źródłowej nie są kopiowane do nowej ścieżki.	Zostanie otwarta strona obrazów stanu aplikacji procesu. Zostanie utworzona nowa ścieżka o nazwie Ścieżka 2. Nowa ścieżka zostanie otwarta, a kursor zostanie umieszczony nad nazwą obrazu stanu, który został skopiowany. Na ekranie zostanie wyświetlony tekst “Tworzenie ścieżki przy użyciu obrazu stanu”.
16	Każda ścieżka zawiera inną najnowszą wersję (bieżącą wersję roboczą). W aplikacji procesu lub bibliotece narzędziowej z wieloma ścieżkami można kopiować zasoby z obrazu stanu jednej ścieżki do najnowszej wersji innej ścieżki. W takim przypadku zmodyfikowane zasoby zastąpią zasoby, które znajdują się już w ścieżce docelowej, a nowe zasoby zostaną dodane.	Nowy obraz stanu zostanie otwarty, a następnie zostanie wyświetlona strona z ustawieniami aplikacji procesu i elementami. Zostanie wyświetlona definicja procesu biznesowego. Zasoby zostaną skopiowane do najnowszej wersji ścieżki Główna. Następnie zostaną wyświetlone strony z listami nowych i zaktualizowanych zasobów oraz tych powodujących konflikt. Na ekranie zostanie wyświetlony tekst “Kopiowanie zasobów do innych ścieżek”.
17	Tutaj przedstawiono hierarchię obrazów stanu aplikacji procesu i ścieżek w repozytorium komponentu Process Center. Na tym diagramie aplikacje procesu A i B mają pojedynczą ścieżkę domyślną, wiele obrazów stanu i jedną najnowszą wersję. Aplikacja procesu C ma dwie ścieżki, z których każda zawiera wiele obrazów stanu i najnowszą wersję. Obrazy stanu i najnowsza wersja w każdej ścieżce są niezależne od siebie.	Zostanie wyświetlony diagram prezentujący relację między najnowszymi wersjami, obrazami stanu i ścieżkami. Diagram przedstawia reprezentacje następujących aplikacji procesu: <ul style="list-style-type: none"> • Aplikacja procesu A, która ma jedną ścieżkę, cztery obrazy stanu i najnowszą wersję. • Aplikacja procesu B, która ma jedną ścieżkę, trzy obrazy stanu i najnowszą wersję. • Aplikacja procesu C, która ma dwie ścieżki. Jedna ścieżka zawiera trzy obrazy stanu i najnowszą wersję, a druga ścieżka - dwa obrazy stanu i najnowszą wersję.

Tabela 5. Scenariusz użycia najnowszych wersji, obrazów stanu i ścieżek

Scena	Audio	Działanie na ekranie
18	Gdy już znane jest przeznaczenie aplikacji procesów i bibliotek narzędziowych, a także sposób użycia najnowszych wersji, obrazów stanu i ścieżek, warto dowiedzieć się, w jaki sposób je wykorzystać. Zostaną prześledzone czynności Roberta, programisty biznesowego w firmie ABC.	Zostanie wyświetlone zdjęcie uśmiechniętego mężczyzny. Zdjęcie ma etykietę "Robert, programista biznesowy w firmie ABC".
19	Robert projektuje proces modelujący sposób zatrudniania nowych pracowników przez jego firmę. Ten proces ma ścieżkę domyślną Główna. Robert wprowadza zmiany w procesie w najnowszej wersji. Aby upewnić się, że proces działa poprawnie, rozpoczyna on sesję odtwarzania, która uruchamia najnowszą wersję na serwerze komponentu Process Center.	<p>Zostanie wyświetlony diagram reprezentujący środowisko programowania aplikacji procesu.</p> <p>W sekcji o nazwie Programowanie zostanie wyświetlony zestaw trzech zagnieżdżonych prostokątów. Zewnętrzny prostokąt reprezentuje repozytorium komponentu Process Center, środkowy prostokąt reprezentuje aplikację procesu, a prostokąt położony najbardziej wewnątrz reprezentuje ścieżkę Główna.</p> <p>Po prawej stronie sekcji Programowanie znajduje się sekcja o nazwie Testowanie. Ta sekcja zawiera pojedynczy prostokąt, który reprezentuje serwer komponentu Process Center.</p> <p>W sekcji Testowanie znajduje się sekcja o nazwie Produkcja. Ta sekcja zawiera pojedynczy prostokąt, który reprezentuje serwer procesów.</p> <p>Aby zaprezentować wprowadzanie przez Roberta zmian w najnowszej wersji, w ścieżce Główna zostanie wyświetlony trójkąt reprezentujący najnowszą wersję. Po uruchomieniu przez Roberta sesji odtwarzania strzałka przedstawia kopiowanie najnowszej wersji na serwer komponentu Process Center.</p>
20	Robert i jego zespół zakończyli projektowanie oraz testowanie aplikacji procesu, która jest gotowa do wdrożenia w środowisku produkcyjnym. W tym celu Robert tworzy obraz stanu aplikacji procesu. Następnie Alicja, administrator, instaluje obraz stanu na serwerze procesów. Firma ABC może teraz używać procesu Roberta w celu przeprowadzenia procesu zatrudniania nowych pracowników.	Aby zaprezentować tworzenie przez Roberta obrazu stanu, obok najnowszej wersji na ścieżce Główna zostanie wyświetlona elipsa reprezentująca obraz stanu. Aby zaprezentować instalowanie przez Alicję obrazu stanu, strzałka przedstawia kopiowanie obrazu stanu na serwer procesów.
21	Później firma ABC ogłasza zmianę marki. W efekcie w przyszłości będzie znana jako firma DEF. Podczas wprowadzania tej zmiany proces zatrudniania utworzony przez Roberta musi zostać zachowany, a jednocześnie musi zostać przygotowana nowa wersja uwzględniająca zaktualizowaną tożsamość firmy. Zespół konserwacyjny wprowadza drobne poprawki w oryginalnym procesie w ścieżce Główna, natomiast Robert i jego zespół tworzą wersję aplikacji zatrudniania dla zmienionej marki na ścieżce Zmiana marki na DEF.	Zostanie wyświetlony dodatkowy obraz stanu w celu zaprezentowania, że w aplikacji procesu wykonano dalsze czynności projektowe. Prostokąt, który reprezentuje ścieżkę Zmiana marki na DEF, zostanie dodany wewnątrz aplikacji procesu. Początkowo ścieżka Zmiana marki na DEF ma jeden obraz stanu i najnowszą wersję, a następnie w ścieżce zostanie dodany kolejny obraz stanu w celu zaprezentowania dalszych czynności projektowych.

Tabela 5. Scenariusz użycia najnowszych wersji, obrazów stanu i ścieżek (kontynuacja)

Scena	Audio	Działanie na ekranie
22	Zarówno zespół konserwacyjny, jak i zespół zmiany marki mogą uruchamiać osobne sesje odtwarzania na serwerze komponentu Process Center. Gdy aplikacje procesów będą gotowe do wdrożenia, każda ścieżka może zainstalować jeden lub więcej obrazów stanu na serwerze procesów.	Strzałki przedstawiają kierunek najnowszych wersji z obu ścieżek działających jednocześnie na serwerze komponentu Process Center. Następnie strzałki przedstawiają obrazy stanu z każdej ścieżki działające jednocześnie na serwerze procesów.

Tabela 6. Podsumowanie

Scena	Audio	Działanie na ekranie
23	W tym filmie wideo przedstawiono aplikacje procesów i biblioteki narzędziowe oraz sposób, w jaki produkt IBM Business Process Manager obsługuje iteracyjne programowanie procesów za pomocą najnowszych wersji, obrazów stanu i ścieżek.	Zostanie wyświetlona wypunktowana lista tematów omówionych w filmie wideo.
24	Aby uzyskać więcej informacji na temat produktu IBM Business Process Manager oraz korzystania z najnowszych wersji, obrazów stanu i ścieżek, należy przejrzeć inne nasze filmy wideo w serwisie YouTube lub zapoznać się poniższymi oficjalnymi zasobami.	Zostanie wyświetlona lista wypunktowana z następującymi zasobami dodatkowymi: <ul style="list-style-type: none"> • Kanał WebSphereEducation w serwisie YouTube • Dokumentacja produktu IBM Business Process Manager 8.5 • IBM developerWorks • IBM Education Assistant

Uruchamianie i debugowanie procesów

Dzięki komponentowi Inspector poszczególni programiści mogą uruchamiać procesy i usługi na serwerze Process Center Server lub zdalnym serwerze wykonawczym Process Server.

Komponent Inspector w produkcie IBM Process Designer jest kluczowym elementem umożliwiającym iteracyjne podejście do programowania procesów. Cały zespół programistyczny może korzystać z komponentu Inspector do demonstrowania bieżącego projektu procesu oraz jego implementacji w sesjach odtwarzania. Sesje odtwarzania ułatwiają uzyskiwanie ważnych informacji od interesariuszy projektu, takich jak kierownictwo, użytkownicy końcowi i analitycy biznesowi. Podejście iteracyjne do programowania procesów zapewnia osiągnięcie zamierzonych celów i zaspokojenie potrzeb wszystkich zaangażowanych osób.

Komponent Inspector w produkcie IBM Process Designer obejmuje kilka narzędzi pozwalających wykonywać wymienione niżej czynności we wszystkich skonfigurowanych środowiskach:

Czynność	Opis
Zarządzanie instancjami procesów	Podczas uruchamiania procesu można wyświetlać wszystkie instancje wykonywane wcześniej i działające w danej chwili na serwerach produktu IBM Business Process Manager w danym środowisku. Zarządzanie działającymi instancjami obejmuje na przykład ich wstrzymywanie, a następnie wznowianie. W przypadku wcześniej wykonywanych instancji zarządzanie polega na filtrowaniu i usuwaniu określonych rekordów.
Wykonywanie krokowe i debugowanie procesu	W trakcie oceniania realizacji procesu przez jego krokowe wykonywanie w każdym momencie widoczny jest krok bieżący wybranej instancji oraz kroki po nim następujące. Obraz procesu w postaci drzewa wraz z kontrolkami, nazywanymi na diagramie procesu znacznikami, ułatwiają przedstawienie bieżącego etapu wykonywania procesu. Tam gdzie stosowane są zmienne, w każdym kroku istnieje także możliwość wyświetlania ich wraz z odpowiadającymi im wartościami.

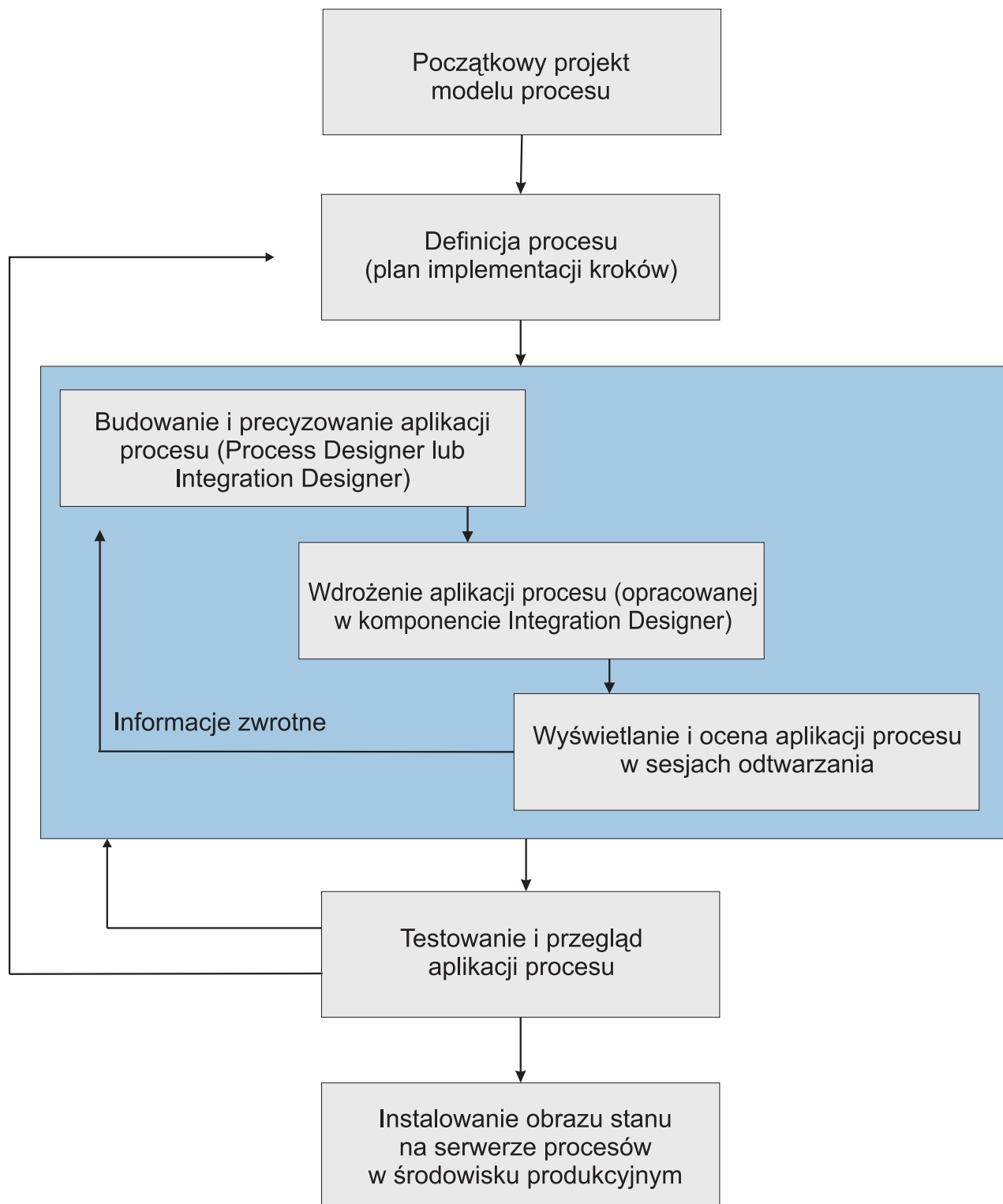
Pracując przy użyciu produktu IBM Integration Designer, można używać komponentu Inspector, jeśli projekt jest powiązany z aplikacją procesu. Użytkownik ma do dyspozycji także inne narzędzia debugowania i testowania. Więcej informacji na temat tych narzędzi produktu Integration Designer zawierają sekcje „Testowanie modułów” i „Korzystanie z debugera integracji na potrzeby określania problemów” dostępne na stronach pokrewnych.

Instalowanie aplikacji procesów i zarządzanie nimi

Cykl życia aplikacji procesu obejmuje instalowanie obrazów stanu, administrowanie nimi oraz wycofywanie ich wdrożenia. Uwagi dotyczące kontroli wersji również stanowią część cyklu życia.

Podczas programowania procesów użytkownik może w pełni wykorzystać podejście iteracyjne obsługiwane przez narzędzia zawarte w produkcie Process Designer. Procesy ewoluują w czasie od etapu początkowego, czyli programowania, przez testowanie, aż do gotowości produkcyjnej. Nawet na etapie produkcyjnym procesy mogą nadal ewoluować z powodu zmieniających się potrzeb. Gotowość do ciągłego udoskonalania procesów jest ważna i ułatwia skuteczne projektowanie od samego początku.

Poniższa ilustracja przedstawia iteracyjne podejście do programowania procesów.



Typowa konfiguracja produktu Business Process Manager obejmuje trzy środowiska obsługujące programowanie i docelowe instalowanie procesów.

Środowisko	Opis
Programistyczne	Budowanie i udoskonalanie aplikacji procesów w produkcie IBM Process Designer. Tworzenie modeli procesów i implementowanie poszczególnych kroków w tych modelach za pomocą komponentu Designer. Przeprowadzanie demonstracji postępu programowania podczas sesji odtwarzania za pomocą komponentu Inspector w celu uzyskiwania szybkiej oceny i udoskonalania prototypu. Instalowanie za pomocą konsoli Process Center Console aplikacji procesów na testowych lub produkcyjnych serwerach procesów.
Testowe	Instalowanie aplikacji procesów na serwerze Process Server w środowisku testowym za pomocą konsoli Process Center Console w celu przeprowadzania formalnych testów zapewniania jakości. Komponent Inspector ułatwia weryfikowanie i rozwiązywanie problemów.
Produkcyjne	Instalowanie aplikacji procesów na serwerze Process Server w środowisku produkcyjnym za pomocą konsoli Process Center Console po rozwiązaniu wszystkich problemów zgłoszonych w trakcie formalnego testowania. Badanie i rozwiązywanie wszystkich problemów zgłoszonych w środowisku produkcyjnym za pomocą komponentu Inspector.

Aby testować lub instalować obraz stanu aplikacji procesu zawierający treść pochodzącą z produktu IBM Business Process Manager Advanced bądź administrować nim, użytkownik lub grupa użytkownika musi być przypisana do administracyjnych ról zabezpieczeń Konfigurujący, Operator *oraz* Wdrażający. Jeśli bieżący użytkownik nie jest przypisany do wszystkich tych ról, należy kliknąć opcję **Użytkownicy i grupy** w Konsoli administracyjnej produktu WebSphere, aby zmodyfikować role użytkownika lub grupy. Patrz temat Role zabezpieczeń produktu IBM Business Process Manager.

Strategie publikowania i instalowania

Aby zapewnić spełnianie standardów jakości obowiązujących w danej organizacji przez implementowane i instalowane aplikacje procesów, należy zdefiniować strategię publikowania i instalowania tych aplikacji. Po zidentyfikowaniu celów i wymagań związanych z publikowaniem i instalowaniem nowych oraz zaktualizowanych aplikacji procesów można zautomatyzować procesy niezbędne przy zatwierdzaniu i uruchamianiu programów.

Użytkownik może na przykład wymagać skierowania procesu do kilku różnych menedżerów należących do różnych struktur raportowania w danej organizacji. Dopiero po zatwierdzeniu przez wszystkich menedżerów nowy lub zaktualizowany proces można zainstalować w środowisku produkcyjnym i przekazać użytkownikom końcowym. W produkcie IBM Business Process Manager Advanced istnieje możliwość utworzenia i zaimplementowania kroków takiego przeglądu, które zapewniają zgodność ze wszystkimi wytycznymi korporacyjnymi i prowadzą do uzyskania zgody wszystkich wymaganych użytkowników. Ostatnim krokiem takiego przeglądu mogłoby być powiadomienie zespołu informatycznego o gotowości zatwierdzonej aplikacji procesu do instalacji.

Tworzenie usług, dostęp do usług oraz ich włączanie

Procesy biznesowe często korzystają z usług, które udostępniają funkcje niezbędne dla danego procesu biznesowego. Te usługi są wyświetlane w diagramie procesu biznesowego jako działanie lub krok. Usługa w komponencie Process Designer może na przykład wywoływać zewnętrzną usługę Web Service lub złożoną i zautomatyzowaną usługę zaprojektowaną w produkcie Integration Designer.

Uzyskiwanie dostępu do usług zewnętrznych dla aplikacji

Ten scenariusz omawia różne sposoby uzyskiwania dostępu do usług zewnętrznych dla aplikacji. Udostępnia również ogólne czynności służące do uzyskiwania dostępu do tych usług zewnętrznych.

Uwaga: Ten scenariusz jest odpowiedni dla produktu IBM Business Process Manager Advanced.

W zintegrowanej aplikacji biznesowej *usługi biznesowe* współdziałają ze sobą w celu realizacji wymaganej funkcji. Usługa biznesowa realizuje powtarzalną funkcję lub czynność, które mają wpływ na osiągnięcie celu biznesowego. Jednak praca związana ze znajdowaniem usługi i nawiązywaniem połączenia z tą usługą nie jest powiązana z funkcją biznesową. Oddzielenie funkcji biznesowej od czynności związanych z zarządzaniem połączeniami usług zapewnia elastyczność rozwiązania.

Interakcja usług rozpoczyna się, gdy *requester usług* wysyła żądanie do *dostawcy usług* w celu wykonania funkcji biznesowej. To żądanie jest wysyłane w formie *komunikatu*, który definiuje funkcję do wykonania. Dostawca usług wykonuje żadaną funkcję i wysyła wynik w postaci komunikatu do requestera usług. Zazwyczaj komunikaty wymagają przetworzenia, aby możliwa była wymiana danych między usługami, a także aby możliwa była implementacja innych niskopoziomowych funkcji informatycznych, które są niezależne od funkcji biznesowych i danych, na przykład: kierowania, konwersji protokołu, transformacji, ponawiania wywołania zakończonym niepowodzeniem oraz dynamicznego wywołania usługi. To przetwarzanie jest nazywane *mediacją*.



W produkcie IBM Integration Designer istnieją dwa typy modułów: moduły (zwane również modułami integracji biznesowej), które są zaprojektowane pod kątem zawierania logiki biznesowej (na przykład procesów biznesowych, reguł biznesowych i biznesowych automatów skończonych) oraz moduły mediacji, które implementują przepływy mediacji. Mimo że niektóre funkcje tych dwóch typów modułów nakładają się, zaleca się odseparowanie logiki biznesowej w modułach biznesowych oraz wykonywanie logiki mediacji przy użyciu modułów mediacji.

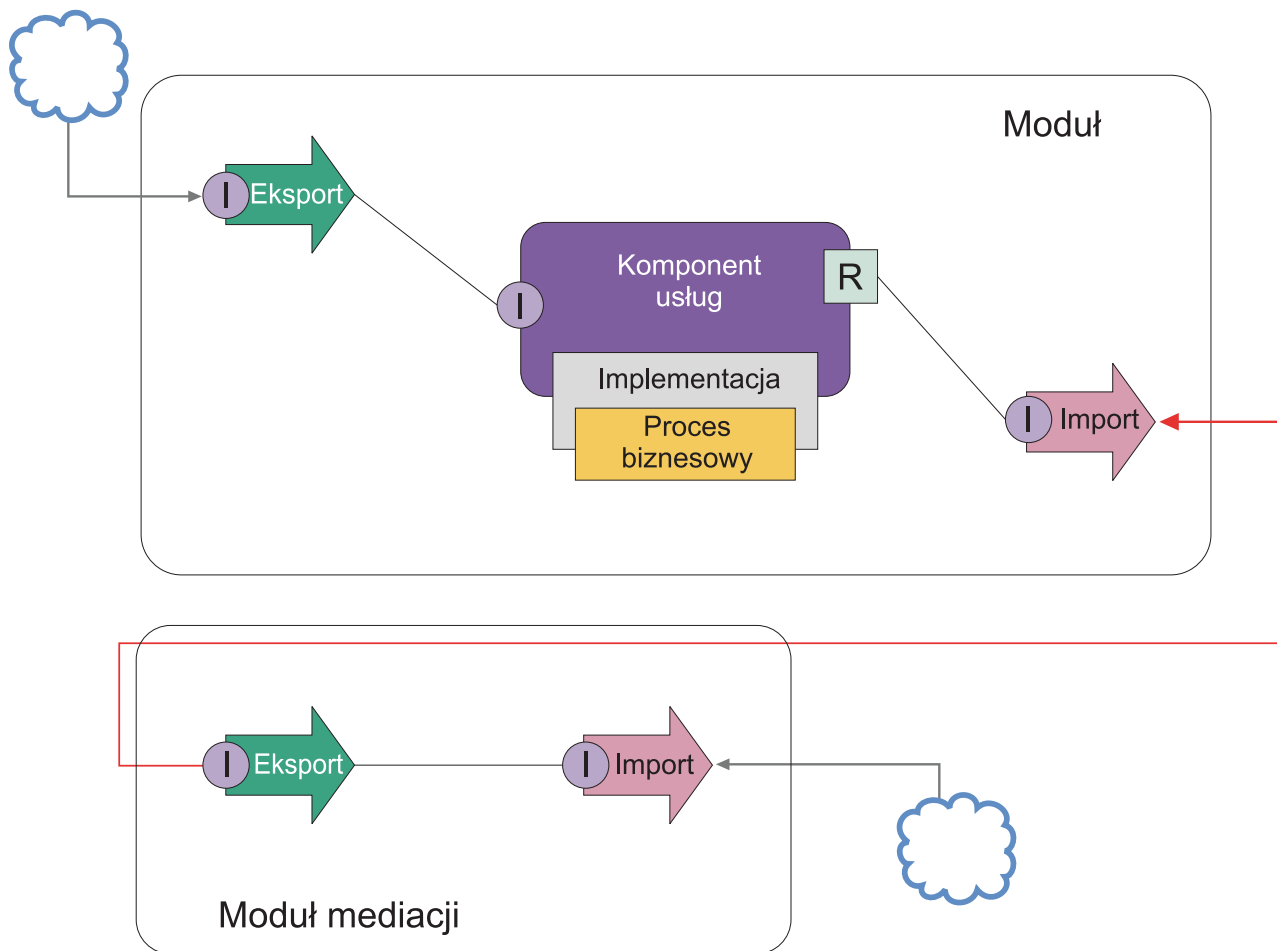
Czasami brakuje jednak wyraźnej granicy między logiką biznesową i logiką mediacji. W takich przypadkach należy uwzględnić liczbę *stanów* lub danych w zmiennych, które będą wymagały przetworzenia między wywołaniami usług. Jeśli wymagane jest niewielkie przetwarzanie stanów lub brak tego przetwarzania, należy rozważyć użycie komponentu przepływu mediacji. Jeśli konieczne jest przechowywanie stanów między wywołaniami usług lub jeśli istnieją dane, które będą musiały być przechowywane w zmiennych i przetwarzane, należy rozważyć użycie komponentu procesu biznesowego. Jeśli na przykład wywoływanych jest kilka usług i rejestrowane są informacje zwracane z każdej usługi, tak aby po wywołaniu wszystkich usług możliwe było przetworzenie zwróconych danych, należy użyć procesu biznesowego, w którym można z łatwością przypisać zwrócone informacje do zmiennych. Innymi słowy, jeśli liczba stanów jest zbyt duża, oznacza to, że została przekroczona granica logiki biznesowej. Wskazówki te są wyjaśnione w poniższych sekcjach.

Istnieje wiele scenariuszy integracji, a żadna odpowiedź nie jest błędna pod względem technicznym. Omawiane wytyczne stanowią sprawdzone procedury, które gwarantują elastyczność i możliwość ponownego wykorzystania. Należy je potraktować jako informacje warte uwzględnienia. Przed zaimplementowaniem tych wzorców w aplikacji integracji biznesowej należy, jak zwykle, dokładnie rozważyć potencjalne korzyści i niedogodności. Poniżej zostanie omówionych kilka przykładowych sytuacji.

Uzyskiwanie dostępu do komponentu SCA

Podstawowym przykładem uzyskiwania dostępu do usługi jest wywołanie innego komponentu SCA przez import, bez konieczności transformacji danych. Nawet w tej sytuacji istnieje możliwość uzyskania dostępu do usługi zewnętrznej z modułu mediacji, zamiast bezpośredniego dostępu z modułu biznesowego. Zapewnia to w przyszłości możliwość elastycznej zmiany punktu końcowego usługi, jakości usług lub zarządzania nimi (na przykład poprzez dodanie rejestrowania) bez wywierania wpływu na komponenty biznesowe będące konsumentami usługi. Taki wzorzec architektury jest nazywany rozdzielaniem kwestii.

Przed podjęciem decyzji o zaimplementowaniu tego wzorca należy rozważyć korzyści płynące z jego użycia oraz potencjalne narzuty wprowadzane przez inny moduł. Jeśli głównym wymaganiem jest uzyskanie elastyczności i planowane jest dokonywanie częstych zmian w usługach, do których uzyskiwany jest dostęp, należy rozważyć wykorzystanie oddzielnego modułu (co pokazano w tym przykładzie). Jeśli najważniejsza jest wydajność, a użytkownik jest gotów na aktualizowanie i ponowne wdrażanie logiki biznesowej, należy rozważyć wykorzystanie pojedynczego modułu.



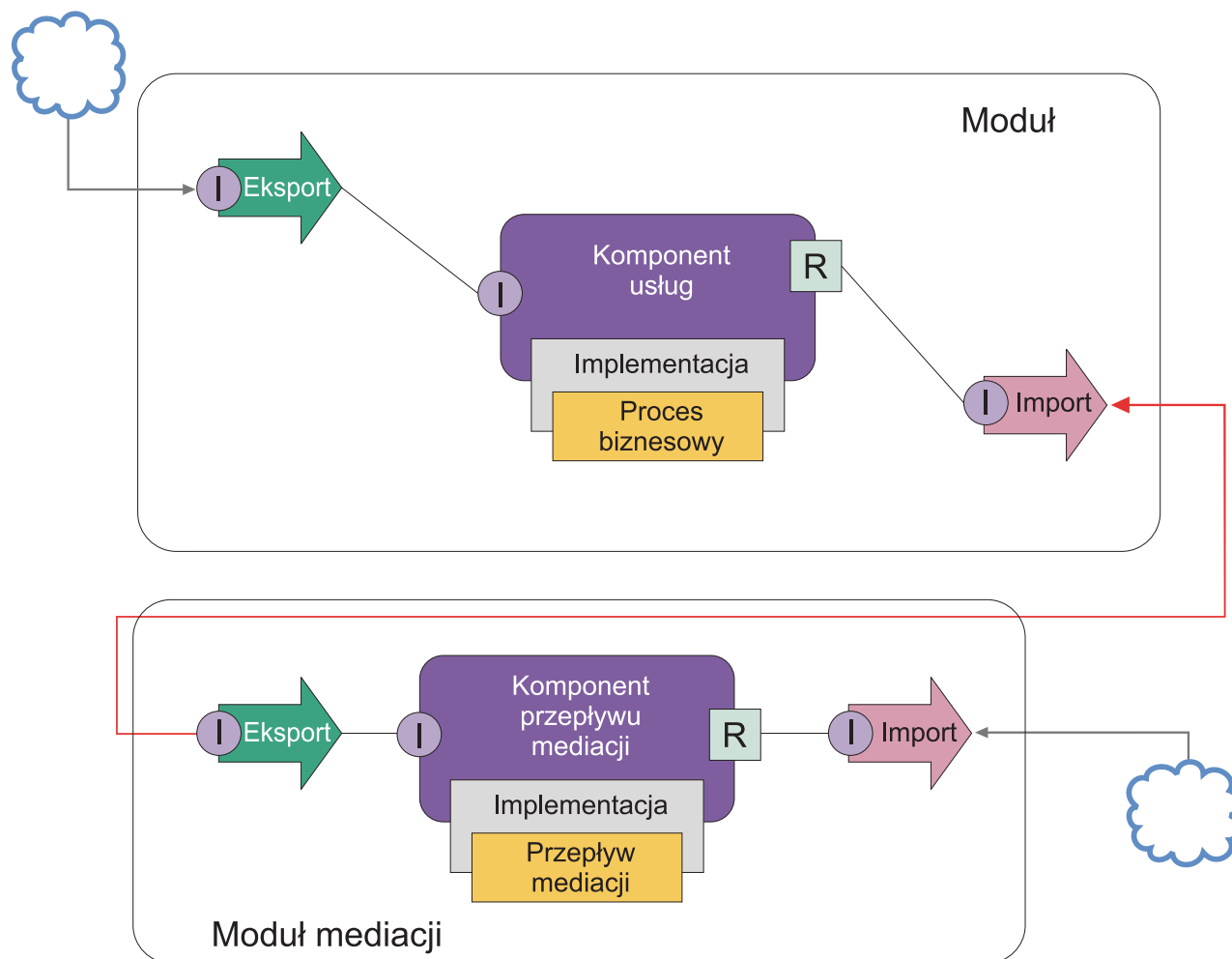
Poniżej przedstawiono ogólne czynności niezbędne do realizacji tego przykładu.

1. Utwórz moduł mediacji. Szczegółowe instrukcje na ten temat zawiera sekcja Tworzenie modułów mediacji.
2. W module mediacji utwórz import z powiązaniem odpowiednim dla zewnętrznej usługi, do której ma być uzyskany dostęp. Szczegółowe instrukcje na ten temat zawiera sekcja Tworzenie importów. Więcej informacji na temat powiązań zawiera sekcja Powiązania.
3. Utwórz eksport i nadaj mu taki sam interfejs, jaki występuje w imporcie. Szczegółowe instrukcje na ten temat zawiera sekcja Tworzenie eksportów.
4. Wygeneruj powiązanie SCA dla eksportu. Szczegółowe instrukcje na ten temat zawiera sekcja Generowanie powiązań SCA.
5. W zespole modułu mediacji połącz eksport z importem. Zapisz moduł mediacji.
6. Utwórz moduł. Szczegółowe instrukcje na ten temat zawiera sekcja Tworzenie modułu na potrzeby usług biznesowych.
7. Dodaj eksport i komponent.
8. W widoku Integracja biznesowa przeciągnij eksport utworzony w module mediacji (krok 4) na zespół modułu. Zostanie utworzony import z takim samym powiązaniem jak eksport.
9. Połącz eksport z komponentem i komponent z importem.
10. Dodaj implementację komponentu. Informacje na temat typów implementacji zawiera sekcja Implementacje.

W późniejszym czasie można dodać do modułu mediacji logikę mediacji, na przykład rejestrowanie lub kierowanie, bez wywierania wpływu na moduł biznesowy.

Dodawanie mediacji

Czasem wywołanie usługi zewnętrznej nie wystarcza i wymagane jest przetworzenie informacji poprzez dodanie modułu mediacji jako elementu pośredniego między requesterem usług i dostawcą usług.



Poniżej przedstawiono niektóre funkcje, które może realizować pośredni przepływ mediacji:

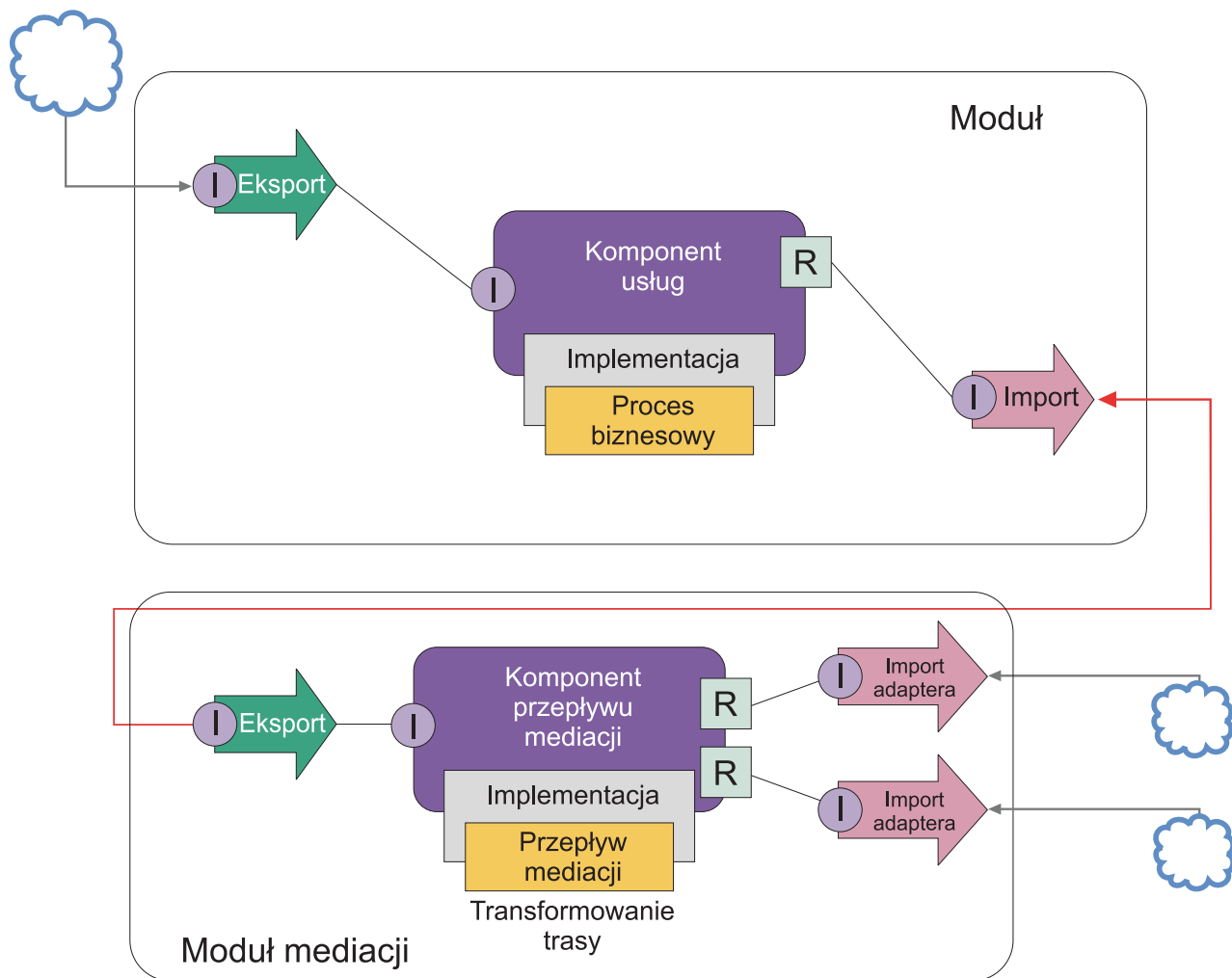
- Ustawianie nagłówek protokołu. Więcej informacji zawiera temat Konwersja protokołu w Centrum informacyjnym produktu WebSphere Enterprise Service Bus.
- Transformowanie interfejsu lub parametru przy użyciu odwzorowania obiektu biznesowego lub operacji podstawowej Odwzorowanie. Więcej informacji na ten temat zawiera sekcja Transformowanie komunikatów.
- Wybór określonej usługi z listy statycznej przy użyciu operacji podstawowej filtru komunikatów. Więcej informacji na ten temat zawiera sekcja Filtr komunikatów.
- Wywoływanie wielu usług w celu zagregowania wyników przy użyciu operacji podstawowych zwielokrotniania wejściowego i zwielokrotniania wyjściowego. Więcej informacji na ten temat zawiera sekcja Agregowanie i rozgłaszanie komunikatów.
- Obsługa niepowodzeń wywołania usługi poprzez ponowną próbę wywołania tej samej usługi lub wywołanie innej usługi przy użyciu operacji podstawowej wywołania usługi. Więcej informacji na ten temat zawiera sekcja Ponowna próba wywołania usługi w przypadku niepowodzenia wywołania.
- Dynamiczne kierowanie poprzez wybór usługi do uruchomienia w czasie wykonywania a nie w czasie integracji. Umożliwia to luźniejsze sprzężenie usług i szybsze reagowanie na zmiany biznesowe. Dzięki temu można dodawać nowe usługi bez zmiany modułów, które zostały wdrożone w środowisku wykonawczym. Dynamiczne kierowanie

zapewnia największe możliwości, jeśli jest używane wraz z rejestrem. Wymaga to użycia operacji podstawowej wyszukiwania punktu końcowego. Więcej informacji na ten temat zawiera sekcja Dynamiczne wybieranie punktów końcowych.

Uzyskiwanie dostępu do systemów EIS

Usługi i artefakty z systemów zewnętrznych można zaimportować do produktu Integration Designer. Kreator pozwala wykryć aplikacje i dane w systemie EIS (Enterprise Information System) oraz umożliwia wygenerowanie usług na podstawie wykrytych aplikacji oraz danych. Generowane są artefakty: interfejsy i obiekty biznesowe, które mogą być używane przez komponenty w module.

Zastosowanie pośredniego modułu mediacji między modulem i hostem umożliwia wielokrotne wykorzystywanie. W poniższym przykładzie przepływ mediacji zastosowano do kierowania do poprawnego hosta oraz do transformowania danych na format wymagany przez host.



Poniżej przedstawiono ogólne czynności dotyczące tego przykładu:

1. Użyj kreatora usług zewnętrznych w celu nawiązania połączenia z hostem. Sposób użycia kreatora usług zewnętrznych do uzyskania dostępu do usług zewnętrznych jest podobny niezależnie od używanego adaptera. Informacje na temat sposobu użycia kreatora usług zewnętrznych zawiera sekcja Wzorzec uzyskiwania dostępu do usług zewnętrznych przy użyciu adapterów.
2. Utwórz moduł. Szczegółowe instrukcje na ten temat zawiera sekcja Tworzenie modułu na potrzeby usług biznesowych.

3. Dodaj eksport, komponent oraz import z powiązaniem SCA. Więcej informacji na ten temat zawiera sekcja Wywoływanie usług.
4. Dodaj interfejs do eksportu i połącz eksport z komponentem.
5. Dodaj implementację komponentu. W implementacji ustaw właściwość określającą, do której usługi hosta uzyskiwany będzie dostęp. Informacje na temat typów implementacji zawiera sekcja Implementacje.
6. Utwórz moduł mediacji z eksportem, który wykorzystuje powiązanie SCA oraz z takim samym interfejsem jak import modułu utworzony w kroku 2.
7. Połącz eksport z komponentem przepływu mediacji.
8. Utwórz import dla każdego hosta, do którego ma zostać uzyskany dostęp, przy użyciu odpowiedniego adaptera wychodzącego z palety edytora składania.
9. Połącz komponent przepływu mediacji z importami.
10. Zaimplementuj komponent przepływu mediacji. Użyj operacji podstawowej filtru komunikatów, aby wybrać import na podstawie właściwości ustawionej w logice biznesowej oraz użyj operacji podstawowej transformacji Odwzorowanie dla każdego importu adaptera.
11. W module wybierz eksport modułu mediacji jako usługę, która ma zostać zaimportowana do modułu. Szczegółowe informacje na ten temat zawiera sekcja Wywoływanie usługi z innego modułu.

W późniejszym czasie istnieje możliwość dokonania zmian takich jak dodanie adaptera lub zmiana adaptera tak, aby wskazywał na inny host. Będzie to miało minimalny wpływ na logikę biznesową.

Uzyskiwanie dostępu do systemów przesyłania komunikatów

Aby moduł SCA (Service Component Architecture) mógł komunikować się z istniejącym klientem przesyłania komunikatów JMS, MQ lub MQ JMS, należy utworzyć interfejsy, obiekty biznesowe oraz powiązania dla importów i eksportów. Patrz sekcja Odwzorowywanie komunikatu na interfejs SCA.

Przepływy mediacji używają komunikatów, które zapewniają nie tylko dostęp do obiektów biznesowych, ale również do kontekstu oraz informacji w nagłówkach. Aby uzyskać dostęp do informacji w nagłówku JMS lub niestandardowej właściwości JMS, należy użyć przepływu mediacji. Jeśli przeprowadzana jest integracja z systemem MQ i potrzebny jest dostęp do informacji w nagłówkach MQ, należy również użyć przepływu mediacji.

Tworzenie lub wywoływanie usługi Web Service

Usługi Web Services są niezależnymi aplikacjami, które realizują funkcje biznesowe - od prostych zapytań po złożone interakcje procesów biznesowych. Użytkownik może wywołać istniejącą usługę Web Service lub utworzyć nową usługę Web Service, tak aby spełniała określone potrzeby. Niniejszy scenariusz opisuje kroki związane z tymi czynnościami oraz wskaże źródła dodatkowych informacji.

Mimo że nie wszystkie usługi będą tworzone od początku w produkcie IBM Integration Designer, niektóre usługi będą w rzeczywistości tworzone w ten sposób. Podczas pracy z edytorem składania i edytorem procesów biznesowych w celu złożenia usług w proces biznesowy użytkownik najprawdopodobniej odkryje, że niektórych usług brakuje. Utworzenie tych brakujących usług przy użyciu narzędzi produktu IBM Integration Designer może okazać się pomocne. Istnieje również możliwość zaistnienia odwrotnej sytuacji - po utworzeniu nowego procesu użytkownik może zdecydować, że użyteczne będzie ujawnienie wszystkich operacji procesu lub podzbioru tych operacji w postaci usługi, którą mogą konsumować inni użytkownicy.

Uwaga: Ten scenariusz dotyczy użytkowników produktu IBM Integration Designer for IBM Process Server.

Istnieje kilka powodów tworzenia usług Web Service przy użyciu produktu IBM Integration Designer:

- Tworzenie usług w produkcie IBM Integration Designer umożliwia implementowanie tych usług przy użyciu reguł biznesowych.
- Programowanie w produkcie IBM Integration Designer umożliwia utworzenie usługi Java™ i ujawnienie jej jako usługi Web Service oraz za pośrednictwem architektury SCA.

- Odzworowywanie interfejsów bez konieczności kodowania jest korzystne. Użytkownik może pobrać wszystkie odzworowania danych z kodu Java, pozostawiając prosty program czarnej skrzynki w języku Java dla programisty Java.
- IBM Integration Designer wyświetla wszystkie usługi i relacje w jednym miejscu.
- Możliwość refaktoryzacji również pomaga w tworzeniu usług Web Service przy użyciu produktu IBM Integration Designer.

Należy pamiętać, że usługi Web Services nie powinny być postrzegane jako rozwiązanie wszystkich problemów z integracją. Jednak podobnie jak w przypadku wszystkich innych technologii oraz podejść do architektury, stosowanie usług Web Services w odpowiednim miejscu i odpowiednim czasie wiąże się z konkretnymi korzyściami.

Eksporty, importy i powiązania

IBM Integration Designer umożliwia importowanie standardowych usług Web Services i wykorzystywanie tych usług w aplikacjach złożonych.

Do tworzenia usług w produkcie IBM Integration Designer służy edytor składania. Tworzenie modułów, modułów mediacji, bibliotek i komponentów wiąże się z wykorzystaniem standardowego procesu. Następnie użytkownik może użyć eksportów, importów i powiązań, aby udostępnić usługi do współużytkowania oraz uzyskać do nich dostęp. Kroki dotyczące tych podstawowych czynności zostały opisane w dalszej części niniejszej sekcji, a odsyłacze prowadzą do bardziej szczegółowych informacji na temat poszczególnych czynności.

Użytkownik może użyć jednego z dwóch typów powiązań na potrzeby usług Web Services: powiązania usługi Web Service lub powiązania HTTP. Powiązanie usługi Web Service udostępnia specyfikację na potrzeby przesyłania komunikatów do usług Web Services i z tych usług. Można je wygenerować automatycznie za pomocą narzędzi. Powiązanie HTTP stanowi standardowy protokół żądanie-odpowiedź realizowany między klientem i serwerem, oparty na definicji protokołu HTTP opublikowanej przez organizację World Wide Web Consortium (W3C). W przypadku używania powiązania HTTP użytkownik musi udostępnić dodatkowe informacje początkowe na temat konfiguracji powiązania.

1. Utwórz eksport, aby opublikować usługę modułu w celu użycia jej przez inne moduły.
2. Wygeneruj powiązanie dla eksportu.
 - Wygeneruj powiązanie usługi Web Service dla eksportu.
 - Wygeneruj powiązanie HTTP dla eksportu.
3. Utwórz import w celu wywołania istniejącej usługi, która nie stanowi części składanego modułu.
 - Wygeneruj powiązanie usługi Web Service dla importu.
 - Wygeneruj powiązanie HTTP dla importu.

Aby wywołać usługę Web Service z poziomu stron JSP, należy zapoznać się z oddzielnym tematem (do którego prowadzi odsyłacz).

Możliwości tworzenia usług Web Services

Podczas otwierania edytora powiązanego z procesem tworzenia usług Web Services może zostać wyświetlone okno potwierdzenia włączenia zawierające następujące informacje:

To działanie wymaga włączenia wdrażania usług Web Services.
Czy włączyć tę możliwość?

IBM Integration Designer udostępnia opcję filtrowania zwaną *możliwościami*. Funkcje i narzędzia są klasyfikowane według możliwości w ustawieniach preferencji. Użytkownik może włączać lub wyłączać kategorie możliwości oraz podzbiory funkcji dowolnych kategorii. Więcej informacji na ten temat zawiera sekcja *Możliwości*.

Dodatkowe informacje o kluczowych pojęciach

Ta sekcja jest punktem wyjścia do zdobywania wiedzy o technologiach stosowanych w produkcie IBM Business Process Manager i wykorzystywanych przez ten produkt.

Scenariusze tworzenia treści

Scenariusze pozwalają zapoznać się z komponentami i produktami z dziedziny zarządzania procesami biznesowymi oraz rozpocząć z nimi pracę.

Kontrola wersji

Cykl życia aplikacji procesu rozpoczyna się od utworzenia aplikacji procesu i jest kontynuowany poprzez cykl aktualizowania, wdrażania, jednoczesnego wdrażania, wycofywania wdrożenia oraz archiwizowania aplikacji procesu. *Kontrola wersji* jest mechanizmem używanym do zarządzania cyklem życia aplikacji procesu poprzez jednoznaczne identyfikowanie poszczególnych wersji aplikacji procesu.

Sposób działania kontroli wersji w produkcie IBM Business Process Manager zależy od tego, jaki element jest wdrażany - czy jest to aplikacja procesu wdrażana z repozytorium komponentu IBM Process Center, czy aplikacja korporacyjna wdrażana bezpośrednio z poziomu produktu IBM Integration Designer.

Aplikacje procesów i biblioteki narzędziowe wdrażane w środowisku wykonawczym z komponentu Process Center są domyślnie objęte kontrolą wersji. W przypadku aplikacji korporacyjnych można wybrać opcję kontroli wersji modułów i bibliotek w produkcie IBM Integration Designer.

Istnieje dodatkowo możliwość tworzenia wersji czynności personelu lub automatu skończonego tak, aby w środowisku wykonawczym mogło jednocześnie istnieć wiele wersji czynności lub automatu skończonego.

Kontrola wersji aplikacji procesów

Kontrola wersji umożliwia środowisku wykonawczemu identyfikowanie obrazów stanu w cyklu życia aplikacji procesów oraz współbieżne uruchamianie wielu obrazów stanu na serwerze Process Server.

Aby zrozumieć, w jaki sposób działa kontrola wersji aplikacji procesów, należy pamiętać, że aplikacja procesu jest kontenerem przechowującym różne aplikacje używane przez aplikację procesu (na przykład modele procesów lub definicje BPD, odwołania do bibliotek narzędziowych, usługi lub ścieżki). Kontrola wersji jest realizowana na poziomie tego kontenera, a nie na poziomie pojedynczych artefaktów. W przypadku aplikacji procesów oznacza to, że kontrola wersji jest realizowana w momencie wykonania obrazu stanu.

Użytkownicy mogą porównywać obrazy stanu, aby określić różnice między wersjami. Jeśli na przykład pewien programista rozwiązał problem występujący w usłudze i wykonał w tym punkcie obraz stanu zawierający ją aplikacji procesów lub biblioteki narzędziowej, a następnie inny programista wprowadził kilka dodatkowych zmian do tej samej usługi i wykonał nowy obraz stanu, to menedżer projektu może porównać te dwa obrazy stanu w celu określenia momentów wprowadzenia zmian oraz ich autorów. Jeśli menedżer projektu uzna, że dodatkowe zmiany wprowadzone do usługi były bez znaczenia, może on przywrócić projekt do stanu z momentu wykonania obrazu stanu zawierającego pierwotną poprawkę.

Istnieje możliwość uruchamiania różnych wersji (obrazów stanu) aplikacji procesu współbieżnie na serwerze. Podczas instalowania nowego obrazu stanu można usunąć oryginalny lub pozostawić go, aby nadal działał.

Kontekst wersji

Każdy obraz stanu ma unikalne metadane umożliwiające identyfikację wersji (zwane kontekstem wersji). Może to być ten identyfikator, ale zalecamy stosowanie trzysegmentowego systemu wersji liczbowych w formacie <wersja główna>.<wersja podrzędna>.<usługa>. Szczegółowy opis tego schematu kontroli wersji zawierają tematy dotyczące konwencji nazewnictwa.

Program IBM Business Process Manager przypisuje każdej aplikacji procesów globalną przestrzeń nazw. Globalna przestrzeń nazw jest albo najnowszą wersją aplikacji procesu, albo konkretnym obrazem stanu aplikacji procesu. Nazwa wersji używana przez serwer może się składać z maksymalnie siedmiu znaków, dlatego przypisana nazwa jest akronimem złożonym ze znaków z nazwy przypisanego obrazu stanu. Akronimy obrazów stanu są takie same, jak nazwy obrazów stanu, jeśli nazwy obrazów stanu są tworzone zgodnie z zaleceniami IBM i nie składają się z więcej niż siedmiu znaków. Na przykład nazwa obrazu stanu 1.0.0 będzie miała akronim 1.0.0, a nazwa obrazu stanu 10.3.0 będzie miała akronim 10.3.0. Zapewni to, że akronim obrazu stanu będzie unikalny w kontekście aplikacji procesów w zasięgu serwera Process Center. Dlatego nie można edytować akronimu obrazu stanu.

Uwagi dotyczące kontroli wersji aplikacji procesów w wielu klastrach

Tę samą wersję aplikacji procesów można zainstalować w wielu klastrach w tej samej komórce. Aby można było rozróżnić wiele instalacji tej samej wersji aplikacji procesów, dla każdej instalacji należy utworzyć obraz stanu i dodać do jej nazwy identyfikator unikalny w całej komórce (na przykład v1.0_kom1_1 i v1.0_kom1_2). Każdy obraz stanu jest nową wersją aplikacji procesów (z perspektywy zarządzania cyklem życia), jednak jego zawartość i funkcje są takie same.

Podczas instalowania aplikacji procesów w klastrze jest wykonywana automatyczna synchronizacja węzłów.

Uwagi dotyczące kontroli wersji bibliotek narzędziowych komponentu Process Designer

Należy pamiętać, że obrazy stanu aplikacji procesów są zazwyczaj wykonywane, kiedy użytkownik jest gotowy do testowania lub instalacji. Z kolei obraz stanu biblioteki narzędziowej jest tworzony, gdy jest ona gotowa do użycia przez aplikacje procesów. Aby w późniejszym czasie zaktualizować bibliotekę narzędziową, należy wykonać kolejny obraz stanu gotowej „najnowszej wersji”, a wtedy właściciele aplikacji procesów i bibliotek narzędziowych będą mogli zdecydować, czy chcą korzystać z tego nowego obrazu stanu.

Kontrola wersji modułów i bibliotek

Jeśli moduł lub biblioteka znajdują się w aplikacji procesu lub bibliotece narzędziowej, przyjmują one cykl życia aplikacji procesu lub biblioteki narzędziowej (wersje, obrazy stanu, ścieżki itp.). Nazwy modułów i bibliotek muszą być unikalne w zasięgu aplikacji procesu lub biblioteki narzędziowej.

Niniejszy temat opisuje kontrolę wersji modułów i bibliotek używanych z aplikacjami procesów. Należy jednak zauważyć, że jeśli moduły są wdrażane bezpośrednio z produktu IBM Integration Designer na serwerze Process Server, można nadal postępować zgodnie z procedurą przypisywania numerów wersji do modułów podczas wdrażania, która została opisana w sekcji “Tworzenie modułów i bibliotek z kontrolą wersji”.

Moduł lub biblioteka, które są powiązane z komponentem IBM Process Center, muszą mieć niezależne biblioteki w tej samej aplikacji procesu lub w zależnej bibliotece narzędziowej.

Poniższa tabela przedstawia wybory, jakich można dokonać w edytorze zależności produktu IBM Integration Designer, jeśli biblioteka jest powiązana z aplikacją procesu lub biblioteką narzędziową:

Tabela 7. Zależności dla modułu, aplikacji procesu lub biblioteki narzędziowej oraz bibliotek globalnych

Zasięg biblioteki	Opis	Możliwa zależność od. . .
Moduł	Kopia tej biblioteki istnieje na serwerze dla każdego modułu korzystającego z tej biblioteki.	Biblioteka w zasięgu modułu może być zależna od wszystkich typów bibliotek.

Tabela 7. Zależności dla modułu, aplikacji procesu lub biblioteki narzędziowej oraz bibliotek globalnych (kontynuacja)

Zasięg biblioteki	Opis	Możliwa zależność od . . .
Aplikacja procesu lub biblioteka narzędziowa	Biblioteka jest współużytkowana przez wszystkie moduły w zasięgu aplikacji procesu lub biblioteki narzędziowej. To ustawienie obowiązuje, jeśli wdrożenie jest realizowane za pośrednictwem komponentu IBM Process Center. Jeśli wdrożenie jest realizowane poza komponentem IBM Process Center, biblioteka jest kopiowana do każdego modułu. Uwaga: Biblioteki utworzone w produkcie IBM Integration Designer 8 mają domyślnie poziom współużytkowania aplikacji procesu lub biblioteki narzędziowej .	Biblioteka tego typu może być zależna tylko od bibliotek globalnych.
Globalne	Biblioteka jest współużytkowana przez wszystkie działające moduły.	Biblioteka globalna może być zależna tylko od innych bibliotek globalnych. Uwaga: W celu wdrożenia biblioteki globalnej należy skonfigurować bibliotekę współużytkowaną produktu WebSphere. Więcej informacji na ten temat zawiera sekcja “Zależności modułów i bibliotek”.

Moduły i biblioteki powiązane z aplikacjami procesów lub bibliotekami narzędziowymi

Nie ma potrzeby kontroli wersji modułów ani bibliotek powiązanych z aplikacjami procesów lub bibliotekami narzędziowymi.

Moduły i biblioteki powiązane z aplikacją procesu lub biblioteką narzędziową nie wymagają kontroli wersji. W rzeczywistości nie można nawet utworzyć wersji modułu lub biblioteki, które są powiązane z aplikacją procesu lub biblioteką narzędziową w edytorze zależności. Moduły i biblioteki powiązane z aplikacją procesu lub biblioteką narzędziową korzystają z obrazów stanu, które są funkcją w komponencie Process Center. Ta funkcja pozwala uzyskać te same wyniki co kontrola wersji.

Biblioteki powiązane z aplikacją procesu lub biblioteką narzędziową nie mają wymaganego numeru wersji w sekcji Biblioteki edytora zależności, ponieważ nie jest potrzebna kontrola wersji.

Konwencje nazewnictwa

Konwencja nazewnictwa służy do rozróżnienia poszczególnych wersji aplikacji procesu w ramach cyklu życia obejmującego aktualizowanie, wdrażanie, jednoczesne wdrażanie, wycofywanie wdrożenia oraz archiwizowanie.

Niniejsza sekcja opisuje konwencje używane do jednoznacznego identyfikowania wersji aplikacji procesu.

Kontekst wersji jest kombinacją akronimów, która jednoznacznie opisuje aplikację procesu lub bibliotekę narzędziową. Każdy typ akronimu ma konwencję nazewnictwa. Akronim może mieć maksymalnie siedem znaków z zestawu znaków [A-Z0-9_], a akronim obrazu stanu może dodatkowo zawierać kropkę.

- Akronim aplikacji procesu jest tworzony podczas tworzenia aplikacji procesu. Może mieć on maksymalnie siedem znaków długości.
- Akronim obrazu stanu jest tworzony automatycznie podczas tworzenia obrazu stanu. Może mieć on maksymalnie siedem znaków długości.

Jeśli nazwa obrazu stanu spełnia kryteria poprawnego akronimu obrazu stanu, nazwa obrazu stanu i jego akronim będą takie same.

Uwaga: Jeśli używana jest funkcja kierowania z uwzględnieniem wersji komponentu przepływu mediacji, obraz stanu należy nazwać tak, aby był zgodny ze schematem <wersja>.<wydanie>.<modyfikacja> (na przykład 1.0.0). Ponieważ akronim obrazu stanu może mieć maksymalnie siedem znaków, wartości cyfrowe mogą mieć

maksymalnie łącznie pięć cyfr (pięć cyfr plus dwie kropki). Dlatego też należy uważać podczas zwiększania wartości pól cyfr, ponieważ wartość jest obcinana do pierwszych siedmiu znaków.

Na przykład akronimem dla nazwy obrazu stanu **11.22.33** będzie akronim **11.22.3**.

- Akronim ścieżki jest generowany automatycznie na podstawie pierwszego znaku każdego słowa w nazwie ścieżki. Jeśli na przykład zostanie utworzona nowa ścieżka o nazwie **Moja nowa ścieżka**, wartością akronimu będzie **Mns**. Domyślną nazwą ścieżki i akronimem ścieżki jest **Main**. Akronim ścieżki jest uwzględniany w kontekście kontroli wersji podczas wdrożenia na serwerze komponentu IBM Process Center, jeśli ten akronim jest inny niż **Main** (Główny).

Definicja procesu biznesowego w aplikacji procesu jest zazwyczaj identyfikowana na podstawie akronimu nazwy aplikacji procesu, akronimu obrazu stanu oraz nazwy definicji procesu biznesowego. O ile to możliwe, należy wybierać unikalne nazwy definicji procesów biznesowych. Jeśli istnieją duplikaty nazw, mogą wystąpić następujące problemy:

- Ujawnianie definicji procesów biznesowych w postaci usług Web Services może być niemożliwe, chyba że zostanie zastosowany pewien rodzaj mediacji.
- Wywołanie definicji procesu biznesowego utworzonej w produkcie IBM Process Designer z poziomu procesu BPEL utworzonego w produkcie IBM Integration Designer może być niemożliwe.

Kontekst wersji jest różny, w zależności od sposobu wdrożenia aplikacji procesu.

Konwencje nazewnictwa dla wdrożeń serwera komponentu Process Center:

Na serwerze komponentu IBM Process Center można wdrożyć obraz stanu aplikacji procesu, a także obraz stanu biblioteki narzędziowej. Można również wdrożyć najnowszą wersję aplikacji procesu lub najnowszą wersję biblioteki narzędziowej (*najnowsza wersja* to bieżąca wersja robocza aplikacji procesu lub biblioteki narzędziowej). Kontekst wersji jest różny, w zależności od typu wdrożenia.

W przypadku aplikacji procesów do jednoznacznego identyfikowania wersji używane są najnowszą wersją aplikacji procesu lub konkretny obraz stanu aplikacji procesu.

Biblioteki narzędziowe można wdrożyć z jedną aplikacją procesu lub kilkoma aplikacjami procesów, ale cykl życia każdej biblioteki narzędziowej jest powiązany z cyklem życia aplikacji procesu. Dla każdej aplikacji procesu na serwerze wdrożona jest jej własna kopia zależnej biblioteki narzędziowej lub zależnych bibliotek narzędziowych. Wdrożona biblioteka narzędziowa nie jest współużytkowana przez aplikacje procesów.

Jeśli ścieżka powiązana z aplikacją procesu ma inną nazwę niż nazwa domyślna **Main** (Główna), akronim ścieżki również stanowi część kontekstu wersji.

Więcej informacji na ten temat zawiera sekcja “Przykłady” na stronie 33 w dalszej części tego tematu.

Obrazy stanu aplikacji procesu

W przypadku wdrażania obrazów stanu aplikacji procesu kontekst wersji jest kombinacją następujących elementów:

- akronimu nazwy aplikacji procesu,
- akronimu ścieżki aplikacji procesu (jeśli używana jest ścieżka inna niż **Main**),
- Akronimu obrazu stanu aplikacji procesu

Autonomiczne biblioteki narzędziowe

W przypadku wdrażania obrazów stanu biblioteki narzędziowej kontekst wersji jest kombinacją następujących elementów:

- akronimu nazwy biblioteki narzędziowej,
- akronimu ścieżki biblioteki narzędziowej (jeśli używana jest ścieżka inna niż **Main**),
- akronimu obrazu stanu biblioteki narzędziowej.

Najnowsze wersje

Najnowsze wersje aplikacji procesu są stosowane podczas testowania iteracyjnego w produkcji Process Designer. Można je wdrażać tylko na serwerach komponentu Process Center.

W przypadku wdrażania najnowszych wersji aplikacji procesu kontekst wersji jest kombinacją następujących elementów:

- akronimu nazwy aplikacji procesu,
- akronimu ścieżki aplikacji procesu (jeśli używana jest ścieżka inna niż **Main**),
- łańcucha „Tip”.

Najnowsze wersje bibliotek narzędziowych są również stosowane podczas testowania iteracyjnego w produkcji Process Designer. Nie są one wdrażane na serwerze produkcyjnym.

W przypadku wdrażania najnowszych wersji biblioteki narzędziowej kontekst wersji jest kombinacją następujących elementów:

- akronimu nazwy biblioteki narzędziowej,
- akronimu ścieżki biblioteki narzędziowej (jeśli używana jest ścieżka inna niż **Main**),
- łańcucha „Tip”.

Przykłady

Zasoby powinny być jednoznacznie nazwane i zewnętrznie identyfikowane przy użyciu kontekstu wersji.

- Poniższa tabela przedstawia przykłady nazw, które są jednoznacznie identyfikowane. W tym przykładzie najnowsza wersja aplikacji procesu używa domyślnej nazwy ścieżki (**Main**):

Tabela 8. Najnowsza wersja aplikacji procesu z domyślną nazwą ścieżki

Typ nazwy	Przykład
Nazwa aplikacji procesu	Aplikacja procesu 1
Akronim nazwy aplikacji procesu	Ap1
Ścieżka aplikacji procesu	Main
Akronim ścieżki aplikacji procesu	"" (jeśli ścieżka to Main)
Obraz stanu aplikacji procesu	
Akronim obrazu stanu aplikacji procesu	Tip

Wszystkie moduły SCA powiązane z tą najnowszą wersją aplikacji procesu zawierają kontekst wersji. Zostało to przedstawione w poniższej tabeli:

Tabela 9. Moduły SCA i pliki EAR z uwzględnieniem wersji

Nazwa modułu SCA	Nazwa z uwzględnieniem wersji	Nazwa pliku EAR / aplikacji z uwzględnieniem wersji
M1	Ap1-Tip-M1	Ap1-Tip-M1.ear
M2	Ap1-Tip-M2	Ap1-Tip-M2.ear

- Poniższa tabela przedstawia przykład najnowszej wersji aplikacji procesu używającej nazwy ścieżki innej niż domyślna:

Tabela 10. Najnowsza wersja aplikacji procesu z nazwą ścieżki inną niż domyślna

Typ nazwy	Przykład
Nazwa aplikacji procesu	Aplikacja procesu 1
Akronim nazwy aplikacji procesu	Ap1

Tabela 10. Najnowsza wersja aplikacji procesu z nazwą ścieżki inną niż domyślna (kontynuacja)

Typ nazwy	Przykład
Ścieżka aplikacji procesu	Sciezka1
Akronim ścieżki aplikacji procesu	S1
Obraz stanu aplikacji procesu	
Akronim obrazu stanu aplikacji procesu	Tip

Wszystkie moduły SCA powiązane z tą najnowszą wersją aplikacji procesu zawierają kontekst wersji. Zostało to przedstawione w poniższej tabeli:

Tabela 11. Moduły SCA i pliki EAR z uwzględnieniem wersji

Nazwa modułu SCA	Nazwa z uwzględnieniem wersji	Nazwa pliku EAR / aplikacji z uwzględnieniem wersji
M1	Ap1-S1-Tip-M1	Ap1-S1-Tip-M1.ear
M2	Ap1-S1-Tip-M2	Ap1-S1-Tip-M2.ear

Podobne konwencje nazewnictwa mają zastosowanie w przypadku wdrożeń zaawansowanych bibliotek narzędziowych i obrazów stanu. Są również stosowane w zaawansowanych obrazach stanu instalowanych na serwerze Process Server.

- Poniższa tabela przedstawia przykłady nazw, które są jednoznacznie identyfikowane. W tym przykładzie obraz stanu aplikacji procesu używa domyślnej nazwy ścieżki (**Main**):

Tabela 12. Obraz stanu aplikacji procesu z domyślną nazwą ścieżki

Typ nazwy	Przykład
Nazwa aplikacji procesu	Aplikacja procesu 1
Akronim nazwy aplikacji procesu	Ap1
Ścieżka aplikacji procesu	Main
Akronim ścieżki aplikacji procesu	"" (jeśli ścieżka to Main)
Obraz stanu aplikacji procesu	Process Shapshot V1
Akronim obrazu stanu aplikacji procesu	PSV1

Wszystkie moduły SCA powiązane z tym obrazem stanu aplikacji procesu zawierają kontekst wersji. Zostało to przedstawione w poniższej tabeli:

Tabela 13. Moduły SCA i pliki EAR z uwzględnieniem wersji

Nazwa modułu SCA	Nazwa z uwzględnieniem wersji	Nazwa pliku EAR / aplikacji z uwzględnieniem wersji
M1	PA1-PSV1-M1	PA1-PSV1-M1.ear
M2	PA1-PSV1-M2	PA1-PSV1-M2.ear

- Poniższa tabela przedstawia przykład obrazu stanu aplikacji procesu używającego nazwy ścieżki innej niż domyślna:

Tabela 14. Obraz stanu aplikacji procesu z nazwą ścieżki inną niż domyślna

Typ nazwy	Przykład
Nazwa aplikacji procesu	Aplikacja procesu 1
Akronim nazwy aplikacji procesu	Ap1
Ścieżka aplikacji procesu	Sciezka1
Akronim ścieżki aplikacji procesu	S1
Obraz stanu aplikacji procesu	Process Snapshot V1

Tabela 14. Obraz stanu aplikacji procesu z nazwą ścieżki inną niż domyślna (kontynuacja)

Typ nazwy	Przykład
Akronim obrazu stanu aplikacji procesu	PSV1

Wszystkie moduły SCA powiązane z tym obrazem stanu aplikacji procesu zawierają kontekst wersji. Zostało to przedstawione w poniższej tabeli:

Tabela 15. Moduły SCA i pliki EAR z uwzględnieniem wersji

Nazwa modułu SCA	Nazwa z uwzględnieniem wersji	Nazwa pliku EAR / aplikacji z uwzględnieniem wersji
M1	PA1-T1-PSV1-M1	PA1-T1-PSV1-M1.ear
M2	PA1-T1-PSV1-M2	PA1-T1-PSV1-M2.ear

Konwencje nazewnictwa dla wdrożeń na serwerze Process Server:

Istnieje możliwość wdrożenia obrazu stanu aplikacji procesu na serwerze Process Server. Do jednoznacznego identyfikowania wersji używany jest akronim obrazu stanu aplikacji procesu.

W przypadku wdrażania obrazów stanu aplikacji procesu kontekst wersji jest kombinacją następujących elementów:

- akronimu nazwy aplikacji procesu,
- akronimu obrazu stanu aplikacji procesu.

Zasoby powinny być jednoznacznie nazwane i zewnętrznie identyfikowane przy użyciu kontekstu wersji. Poniższa tabela przedstawia przykłady nazw, które są jednoznacznie identyfikowane:

Tabela 16. Przykłady nazw i akronimów

Typ nazwy	Przykład
Nazwa aplikacji procesu	Aplikacja procesu 1
Akronim nazwy aplikacji procesu	Ap1
Obraz stanu aplikacji procesu	1.0.0
Akronim obrazu stanu aplikacji procesu	1.0.0

Elementem identyfikowania zasobu takiego jak moduł lub biblioteka jest kontekst wersji.

Poniższa tabela przedstawia przykład dwóch modułów oraz uwzględniania kontekstu wersji w powiązanych plikach EAR:

Tabela 17. Moduły SCA i pliki EAR z uwzględnieniem wersji

Nazwa modułu SCA	Nazwa z uwzględnieniem wersji	Nazwa pliku EAR / aplikacji z uwzględnieniem wersji
M1	Ap1-1.0.0-M1	Ap1-1.0.0-M1.ear
M2	Ap1-1.0.0-M2	Ap1-1.0.0-M2.ear

Poniższa tabela przedstawia przykład dwóch bibliotek w zasięgu aplikacji procesu oraz uwzględniania kontekstu wersji w powiązanych plikach JAR:

Tabela 18. Biblioteki w zasięgu aplikacji procesu oraz pliki JAR z uwzględnieniem wersji

Nazwa biblioteki SCA w zasięgu aplikacji procesu	Nazwa z uwzględnieniem wersji	Nazwa pliku JAR z uwzględnieniem wersji
Lib1	Ap1-1.0.0-Lib1	Ap1-1.0.0-Lib1.jar
Lib2	Ap1-1.0.0-Lib2	Ap1-1.0.0-Lib2.jar

Powiązania z uwzględnieniem wersji

Aplikacje procesów mogą zawierać moduły SCA zawierające powiązania importu i eksportu. Podczas jednoczesnego wdrażania aplikacji powiązania dla każdej wersji aplikacji muszą być unikalne. Niektóre powiązania są automatycznie aktualizowane podczas wdrażania tak, aby zagwarantować unikalność dla poszczególnych wersji. W innych przypadkach użytkownik musi samodzielnie zaktualizować powiązanie po wdrożeniu, aby zagwarantować jego unikalność.

Powiązanie z uwzględnieniem wersji jest w zasięgu konkretnej wersji aplikacji procesu, co gwarantuje jego unikalność dla poszczególnych aplikacji procesów. Poniższe sekcje opisują powiązania, które są automatycznie aktualizowane w celu uwzględnienia wersji, a także wszystkie działania, jakie powinien podjąć użytkownik w czasie wykonywania, jeśli powiązanie nie uwzględnia wersji. Informacje dotyczące kwestii, które powinny być uwzględnione podczas tworzenia modułów, zawiera sekcja “Uwagi na temat używania powiązań”.

SCA

Jeśli powiązania importu i eksportu modułu są zdefiniowane w zasięgu tej samej aplikacji procesu, nazwa elementu docelowego powiązania SCA jest automatycznie zmieniana tak, aby uwzględniała wersję podczas wdrażania.

Jeśli powiązania nie są zdefiniowane w zasięgu tej samej aplikacji procesu, rejestrowany jest komunikat informacyjny. Użytkownik musi zmodyfikować powiązanie importu po wdrożeniu, zmieniając adres docelowy punktu końcowego. Do zmiany adresu docelowego punktu końcowego można użyć Konsoli administracyjnej.

Usługa Web Service (JAX-WS lub JAX-RPC)

Nazwa docelowego adresu punktu końcowego dla powiązania usługi Web Service jest automatycznie zmieniana tak, aby uwzględniała wersje, jeśli spełnione są wszystkie następujące warunki:

- Jeśli użytkownik zastosował domyślną konwencję nazewnictwa dla adresu:
http://ip:port/nazwa_modułuWeb/sca/nazwa_eksportu
- Jeśli adres punktu końcowego jest adresem SOAP/HTTP.
- Jeśli powiązania importu i eksportu dla modułu są zdefiniowane w zasięgu tej samej aplikacji procesu.

Jeśli te warunki nie są spełnione, rejestrowany jest komunikat informacyjny. Działanie, które należy podjąć, zależy od tego w jaki sposób wdrażana jest aplikacja procesu:

- Jeśli aplikacja procesu jest wdrażana jednocześnie, należy ręcznie zmienić nazwę adresu URL punktu końcowego SOAP/HTTP lub nazwę kolejki docelowej SOAP/JMS tak, aby były one unikalne dla poszczególnych wersji aplikacji procesu. Do zmiany adresu docelowego punktu końcowego po wdrożeniu można użyć Konsoli administracyjnej.
- Jeśli wdrażana jest tylko pojedyncza wersja aplikacji procesu, można zignorować ten komunikat.

W przypadku jednoczesnego wdrażania obrazów stanu powiązania usługi Web Service SOAP/JMS działanie, które należy podjąć, zależy od tego w jaki sposób wdrażana jest aplikacja procesu:

- Jeśli import i eksport docelowy znajdują się w tej samej aplikacji procesu, przed opublikowaniem aplikacji procesu w komponencie Process Center i utworzeniem obrazu stanu wykonaj następujące kroki:
 1. Zmień adres URL punktu końcowego dla eksportu. Upewnij się, że miejsce docelowe i fabryka połączeń są unikalne.

2. Zmień adres URL punktu końcowego dla importu tak, aby był on taki sam jak określony dla eksportu w poprzednim kroku.
- Jeśli import i eksport docelowy znajdują się w różnych aplikacjach procesów, wykonaj następujące kroki:
 1. Zmień adres URL punktu końcowego dla eksportu. Upewnij się, że miejsce docelowe i fabryka połączeń są unikalne.
 2. Opublikuj aplikację procesu w komponencie Process Center.
 3. Utwórz obraz stanu.
 4. Wdróż aplikację procesu na serwerze Process Server.
 5. Użyj Konsoli administracyjnej produktu WebSphere w celu zmiany adresu URL punktu końcowego odpowiedniego importu tak, aby był on taki sam jak określony dla eksportu.

HTTP

Nazwa adresu punktu końcowego dla powiązania HTTP jest automatycznie zmieniana tak, aby uwzględniała wersje, jeśli spełnione są wszystkie następujące warunki:

- Jeśli użytkownik zastosował domyślną konwencję nazewnictwa dla adresu:
`http(s)://ip:port/nazwa_modułuWeb/ścieżka_kontekstu_w_eksporcie`
- Jeśli powiązania importu i eksportu dla modułu są zdefiniowane w zasięgu tej samej aplikacji procesu.

Jeśli te warunki nie są spełnione, rejestrowany jest komunikat informacyjny. Działanie, które należy podjąć, zależy od tego w jaki sposób wdrażana jest aplikacja procesu:

- Jeśli aplikacja procesu jest wdrażana jednocześnie, należy ręcznie zmienić nazwę adresu URL punktu końcowego tak, aby była ona unikalna dla poszczególnych wersji aplikacji procesu. Do zmiany adresu docelowego punktu końcowego po wdrożeniu można użyć Konsoli administracyjnej.
- Jeśli wdrażana jest tylko pojedyncza wersja aplikacji procesu, można zignorować ten komunikat.

Powiązania JMS i ogólne powiązania JMS

Powiązania JMS i ogólne powiązania JMS wygenerowane przez system automatycznie uwzględniają wersje.

Uwaga: W przypadku powiązań JMS i ogólnych powiązań JMS zdefiniowanych przez użytkownika, podczas wdrażania nazwy nie są automatycznie zmieniane w celu uwzględniania wersji. Jeśli powiązanie jest zdefiniowane przez użytkownika, należy zmienić nazwy następujących atrybutów tak, aby były unikalne dla poszczególnych wersji aplikacji procesu:

- konfiguracja punktu końcowego,
- kolejka miejsca docelowego odbioru,
- nazwa portu nasłuchiwania (jeśli jest zdefiniowana).

Jeśli punkt końcowy modułu docelowego ulegnie zmianie, należy ustawić zgodne miejsce docelowe Send.

Powiązania MQ/JMS i MQ

Podczas wdrażania powiązań typu MQ/JMS i MQ nazwy nie są automatycznie zmieniane w celu uwzględniania wersji.

Należy zmienić nazwy następujących atrybutów tak, aby były unikalne dla poszczególnych wersji aplikacji procesu:

- konfiguracja punktu końcowego,
- kolejka miejsca docelowego odbioru.

Jeśli punkt końcowy modułu docelowego ulegnie zmianie, należy ustawić zgodne miejsce docelowe Send.

EJB

Podczas wdrażania powiązań typu EJB nazwy nie są automatycznie zmieniane w celu uwzględnienia wersji.

Należy zmienić atrybut nazw JNDI tak, aby był unikalny dla poszczególnych wersji aplikacji procesu.

Należy zauważyć, że aplikacje klienckie również muszą zostać zaktualizowane tak, aby używały nowych nazw JNDI.

EIS

Nazwa adaptera zasobu jest automatycznie zmieniana podczas wdrażania w celu uwzględnienia wersji, pod warunkiem że nie została zmodyfikowana domyślna nazwa zasobu (*nazwa_moduluApp:Adapter Description*).

Jeśli domyślna nazwa zasobu została zmodyfikowana, nazwy adapterów zasobów muszą być unikalne dla poszczególnych wersji aplikacji procesu.

Jeśli nazwy adaptera zasobu nie są unikalne, podczas wdrażania nie jest rejestrowany żaden komunikat informacyjny, który ostrzega użytkownika. Nazwy adapterów zasobów można zmienić ręcznie po wdrożeniu przy użyciu Konsoli administracyjnej.

Wywoływanie dynamiczne z uwzględnieniem wersji

Komponenty przepływu mediacji można skonfigurować tak, aby kierowały komunikaty do punktów końcowych określanych dynamicznie w czasie wykonywania. Podczas tworzenia modułu mediacji należy skonfigurować wyszukiwanie punktów końcowych tak, aby używane było kierowanie z uwzględnieniem wersji.

Jeśli na potrzeby obrazu stanu używany jest styl IBM_VRM (<wersja>.<wydanie>.<modyfikacja>), można wyeksportować plik EAR aplikacji procesu do produktu WebSphere Service Registry and Repository (WSRR). Podczas tworzenia modułu mediacji należy następnie skonfigurować wyszukiwanie punktów końcowych tak, aby używane było kierowanie z uwzględnieniem wersji. Na przykład w polu **Dopasowanie strategii** należy wybrać opcję **Punkt końcowy powrotu odpowiadający ostatniej zgodnej wersji usług SCA opartych na module**, a w polu **Typ powiązania** należy wybrać opcję **SCA**.

Kolejne wersje aplikacji procesu będą wdrażane na serwerze i publikowane w repozytorium WSRR, a wyszukiwanie punktów końcowych modułu mediacji będzie dynamicznie wywoływało ostatnią zgodną wersję punktu końcowego usługi.

Należy zauważyć, że alternatywnym wyjściem jest ustawienie elementu docelowego w polu SMOHeader, a wartość może być przenoszona przez komunikat żądania.

Wdrażanie aplikacji procesów z modułami i projektami Java

Aplikacje procesów mogą zawierać niestandardowe moduły Java EE i projekty Java. Podczas jednoczesnego wdrażania aplikacji niestandardowy moduł Java EE dla każdej wersji aplikacji musi być unikalny.

Należy zauważyć, że niestandardowe moduły Java EE i projekty Java są wdrażane na serwerze, jeśli są wdrażane z modułem SCA, który ma zadeklarowaną zależność od tych modułów lub projektów. Jeśli podczas deklarowania zależności nie zostanie wybrana opcja **Wdróż z modułem** (opcja domyślna), należy ręcznie wdrożyć moduł lub projekt.

Wdrażanie aplikacji procesów z regułami biznesowymi i selektorami

Jeśli użytkownik wdraża wiele wersji aplikacji procesu, która zawiera regułę biznesową lub komponent selektora, należy wiedzieć w jaki sposób poszczególne wersje używają powiązanych metadanych.

Metadane dynamiczne dla reguły biznesowej lub komponentu selektora są definiowane w czasie wykonywania na podstawie nazwy komponentu, docelowej przestrzeni nazw komponentu oraz typu komponentu. Jeśli dwie wersje

aplikacji procesu zawierającej regułę biznesową lub selektor zostaną wdrożone do tego samego środowiska wykonawczego, będą one współużytkowały te same metadane logiki reguły (reguły biznesowej) lub kierowania (selektora).

Aby każda wersja reguły biznesowej lub komponentu selektora w aplikacji procesu mogła używać własnych metadanych dynamicznych (logiki reguły lub kierowania), należy przeprowadzić refaktoryzację docelowej przestrzeni nazw tak, aby była ona unikalna dla każdej wersji aplikacji procesu.

Obiekty konfiguracyjne

Przy użyciu komend AdminConfig narzędzia administracyjnego wiersza komend produktu WebSphere (wsadmin) można uzyskiwać dostęp do bazy danych i właściwości zabezpieczeń produktu IBM Business Process Manager, a także modyfikować je.

Termin *obiekt konfiguracyjny* oznacza obiekt, do którego dostęp jest uzyskiwany za pośrednictwem komend wsadmin AdminConfig. Aby uzyskać więcej informacji, patrz Właściwości konfiguracji zabezpieczeń.

Architektura wdrażania

Architektura wdrażania produktu IBM Business Process Manager składa się z procesów oprogramowania zwanych serwerami, jednostek topologii zwanych węzłami i komórkami oraz repozytorium konfiguracji służącego do przechowywania informacji konfiguracyjnych.

Komórki

W produkcie IBM Business Process Manager *komórki* są grupami logicznymi zawierającymi co najmniej jeden węzeł w sieci rozproszonej.

Komórka jest pojęciem konfiguracyjnym - przeznaczoną dla administratorów metodą tworzenia powiązań logicznych między węzłami. Administratorzy definiują węzły składające się na komórkę zgodnie z konkretnymi kryteriami, które są odpowiednie w danym środowisku organizacyjnym.

Dane konfiguracji administracyjnej są zapisywane w plikach XML. Komórka zachowuje główne pliki konfiguracyjne dla każdego serwera w każdym węźle komórki. Każdy węzeł i każdy serwer mają również własne lokalne pliki konfiguracyjne. Zmiany w węźle lokalnym lub w pliku konfiguracyjnym serwera są tymczasowe, jeśli serwer należy do komórki. Zmiany lokalne nadpisują konfiguracje komórki w czasie, kiedy obowiązują. Zmiany plików konfiguracyjnych głównego serwera i głównego węzła dokonane na poziomie komórki zastępują jakiekolwiek zmiany tymczasowe dokonane na poziomie węzła. Jest to realizowane, kiedy dokumenty konfiguracji komórki są synchronizowane z węzłami. Synchronizacja jest wykonywana w ramach wyznaczonych zdarzeń, na przykład podczas uruchamiania serwera.

Serwery

Serwery udostępniają podstawową funkcjonalność produktu IBM Business Process Manager. Serwery Process Server rozszerzają możliwość obsługi modułów SCA (Service Component Architecture) przez serwer aplikacji. Inne serwery (menedżery wdrażania oraz agenty węzłów) są używane do zarządzania serwerami procesów.

Serwer procesów może być *serwerem autonomicznym* lub *serwerem zarządzanym*. Serwer zarządzany może opcjonalnie być elementem *klastra*. Kolekcja serwerów zarządzanych, klastrów serwerów, a także innego oprogramowania pośredniego jest nazywana *środowiskiem wdrażania*. W środowisku wdrażania każdy zarządzany serwer lub klastr jest skonfigurowany pod kątem realizowania określonej funkcji w środowisku wdrażania (na przykład hosta miejsca docelowego, hosta modułu aplikacji lub serwera infrastruktury Common Event Infrastructure). Serwer autonomiczny jest konfigurowany na potrzeby udostępniania wszystkich wymaganych funkcji.

Serwery udostępniają środowisko wykonawcze dla modułów SCA, dla zasobów używanych przez te moduły (źródeł danych, specyfikacji aktywowania i miejsc docelowych JMS) oraz dla zasobów dostarczanych przez IBM (miejsc docelowych komunikatów, kontenerów produktu Business Process Choreographer i serwerów infrastruktury Common Event Infrastructure).

Agent węzła to agent administracyjny reprezentujący węzeł w systemie użytkownika, który zarządza serwerami w tym węźle. Agenty węzłów monitorują serwery na hoście oraz kierują żądania administracyjne do serwerów. Agent węzła zostaje utworzony po stowarzyszeniu węzła z menedżerem wdrażania.

Menedżer wdrażania to agent administracyjny, który udostępnia centralny widok zarządzania dla wielu serwerów i klastrów.

Serwer autonomiczny jest definiowany przy użyciu profilu autonomicznego. Menedżer wdrażania jest definiowany przy użyciu profilu menedżera wdrażania. Serwery zarządzane są tworzone w *węźle zarządzanym*, który jest definiowany przy użyciu profilu węzła zarządzanego.

Serwery autonomiczne:

Serwer autonomiczny udostępnia środowisko do wdrażania modułów SCA w jednym procesie serwera. Ten proces serwera zawiera co najmniej Konsolę administracyjną, miejsce docelowe wdrażania, obsługę przesyłania komunikatów, menedżer reguł procesów biznesowych oraz serwer infrastruktury CEI (Common Event Infrastructure).

Serwer autonomiczny można łatwo skonfigurować. Udostępnia on konsolę Szybki start, za pomocą której można go uruchamiać i zatrzymywać oraz otwierać galerię przykładów i Konsolę administracyjną. Jeśli zostaną zainstalowane przykłady dla produktu IBM Business Process Manager, a następnie zostanie otwarta galeria przykładów, rozwiązanie przykładowe zostanie wdrożone na serwerze autonomicznym. Przy użyciu Konsoli administracyjnej można przeglądać zasoby użyte w tym przykładzie.

Na serwerze autonomicznym można wdrażać własne rozwiązania, ale nie udostępnia on takiej mocy obliczeniowej, skalowalności ani stabilności, które są wymagane w środowisku produkcyjnym. W przypadku środowisk produkcyjnych lepszym rozwiązaniem jest użycie środowiska wdrożenia sieciowego.

Istnieje możliwość rozpoczęcia pracy z serwerem autonomicznym, a następnie dodanie go do środowiska wdrożenia sieciowego przez stowarzyszenie go w komórce menedżera wdrażania, *o ile z tą komórką nie zostały stowarzyszone żadne inne węzły*. W jednej komórce nie można stowarzyszyć wielu serwerów autonomicznych. Aby stowarzyszyć serwer autonomiczny, należy użyć Konsoli administracyjnej menedżera wdrażania lub komendy **addNode**. Podczas stowarzyszania przy użyciu komendy **addNode** serwer autonomiczny nie może działać.

Serwer autonomiczny jest definiowany przez profil serwera autonomicznego.

Klastry:

Klastry to wspólnie zarządzane grupy serwerów (także pod względem podziału obciążenia).

Klaster może zawierać węzły lub pojedyncze serwery aplikacji. Węzeł jest zazwyczaj fizycznym systemem komputerowym z odrębnym adresem IP, na którym działa co najmniej jeden serwer aplikacji. Klastry mogą być grupowane w ramach konfiguracji komórki, co umożliwia wzajemne logiczne powiązanie wielu serwerów i klastrów z różnymi konfiguracjami i aplikacjami, w zależności od wyboru administratora oraz od tego, czy określone rozwiązanie jest odpowiednie w danym środowisku organizacyjnym.

Klastry odpowiadają za równoważenie obciążenia serwerów. Serwery stanowiące część klastra są nazywane elementami klastra. Podczas instalowania aplikacji w klastrze aplikacja ta jest automatycznie instalowana w każdym elemencie klastra.

Ponieważ każdy element klastra zawiera te same aplikacje, istnieje możliwość rozdzielania zadań klienta w zależności od mocy obliczeniowej poszczególnych komputerów. Jest to realizowane poprzez przypisywanie wag do poszczególnych serwerów.

Przypisywanie wag do serwerów w klastrze pozwala zwiększyć wydajność i usprawnić przełączanie awaryjne. Zadania są przypisywane do serwerów, które mają moc obliczeniową wymaganą do wykonania operacji w ramach zadania. Jeśli jeden serwer jest niedostępny i nie może wykonać zadania, zadanie to jest przypisywane do innego elementu klastra. Ta

możliwość zmiany przypisania wiąże się z oczywistymi korzyściami w porównaniu z sytuacją, w której działa pojedynczy serwer aplikacji, ponieważ pojedynczy serwer może zostać przeciążony w przypadku zbyt dużej liczby żądań.

Profile

Profil definiuje unikalne środowisko wykonawcze z oddzielnymi plikami komend, plikami konfiguracyjnymi i plikami dzienników. Profile definiują trzy różne typy środowisk w systemach produktu IBM Business Process Manager: serwer autonomiczny, menedżer wdrażania oraz węzeł zarządzany.

Profile pozwalają na uzyskanie w systemie więcej niż jednego środowiska wykonawczego bez konieczności instalowania wielu kopii plików binarnych produktu IBM Business Process Manager.

Należy użyć produktu Program narzędziowy wiersza komend BPMConfig, aby utworzyć profile produktu IBM BPM. Program narzędziowy wiersza komend **manageprofiles** lub jego graficzny interfejs użytkownika, czyli narzędzie Profile Management Tool (PMT), mogą być używane jako alternatywne sposoby tworzenia profili menedżera wdrażania lub profili węzła zarządzanego. Narzędzie PMT nie jest już obsługiwane w przypadku tworzenia profilu autonomicznego.

Uwaga: W przypadku platform rozproszonych każdy profil ma unikalną nazwę. W systemie z/OS wszystkie profile mają nazwę "default". W systemie z/OS nie można zmieniać nazwy, edytować, kopiować ani usuwać profili.

Typy profili

Następujące typy profili produktu IBM BPM są dostępne w produkcie IBM Business Process Manager 8.5:

Profil autonomiczny produktu IBM BPM

Profil autonomiczny definiuje serwery autonomiczne o możliwościach i funkcjach specyficznych dla konfiguracji produktu IBM BPM Express. Możliwe jest utworzenie profilu autonomicznego za pomocą szablonu profilu BPM/BpmServer, który jest obsługiwany tylko w produkcie IBM BPM Express.

Profil menedżera wdrażania produktu IBM BPM

Profil menedżera wdrażania definiuje menedżer wdrażania, który udostępnia pojedynczy interfejs administracyjny logicznej grupie serwerów na jednej lub większej liczbie stacji roboczych. Możliwe jest utworzenie profilu menedżera wdrażania za pomocą szablonu profilu BPM/BpmDmgr, który jest obsługiwany tylko w produktach IBM BPM Standard i IBM BPM Advanced.

Profil węzła zarządzanego produktu IBM BPM

Profil węzła zarządzanego definiuje węzeł zarządzany w przypadku stowarzyszenia węzła z menedżerem wdrażania. Możliwe jest utworzenie profilu węzła zarządzanego za pomocą szablonu profilu BPM/BpmNode, który jest obsługiwany tylko w produktach IBM BPM Standard i IBM BPM Advanced.

Dla każdej konfiguracji produktu IBM BPM dostępne są specyficzne typy profili.

Tabela 19. Dostępne typy profili produktu IBM BPM

Konfiguracja produktu IBM BPM	Typy profili		
	Autonomiczne	Menedżer wdrażania	Węzeł zarządzany
IBM BPMPExpress	Tak	Nie	Nie
IBM BPM Standard	Nie	Tak	Tak
IBM BPM Advanced	Nie	Tak	Tak
Środowisko testów jednostkowych dla produktu IBM Integration Designer	Tak	Opcjonalne	Opcjonalne

Katalog profilu

Dla każdego profilu w systemie istnieje osobny katalog zawierający wszystkie pliki tego profilu. Położenie katalogu profilu należy wskazać podczas tworzenia profilu. Domyślnie znajduje się on w podkatalogu `profiles` w katalogu, w którym zainstalowano produkt IBM Business Process Manager. Na przykład: profil `Dmgr` znajduje się w katalogu `C:\Program Files\IBM\WebSphere\ProcServer\profiles\Dmgr`.

Profil domyślny

Pierwszym profilem tworzonym w ramach instalacji produktu IBM Business Process Manager jest *profil domyślny*. Profil domyślny jest domyślnym obiektem docelowym komend wykonywanych z poziomu podkatalogu `bin` w katalogu, w którym zainstalowano produkt IBM Business Process Manager. Jeśli w systemie istnieje tylko jeden profil, wszystkie komendy są wykonywane względem tego profilu. Po utworzeniu dodatkowego profilu można określić, że ma to być profil domyślny.

Uwaga: Profilem domyślnym nie musi być profil o nazwie “default”.

Rozszerzanie profili

Jeśli istnieje już profil menedżera wdrażania, profil węzła zarządzanego lub profil serwera autonomicznego utworzony dla produktu WebSphere Application Server Network Deployment, profil można *rozszerzyć* o obsługę produktu IBM Business Process Manager, uzupełniając w ten sposób istniejącą funkcję. Przed rozszerzeniem profilu należy zainstalować produkt IBM Business Process Manager. Następnie za pomocą programu narzędziowego wiersza komend **manageprofiles** należy rozszerzyć profil autonomiczny albo użyć narzędzia Profile Management Tool lub programu narzędziowego wiersza komend **manageprofiles** do rozszerzenia profilu menedżera wdrażania lub profilu węzła zarządzanego.

Ważne: W środowisku wdrożenia sieciowego należy w pierwszej kolejności rozszerzyć profil menedżera wdrażania, a następnie rozszerzyć profile węzła zarządzanego.

Ograniczenie: Nie można rozszerzyć profilu autonomicznego lub profilu menedżera wdrażania, w przypadku którego zmieniono domyślny rejestr użytkowników menedżera VMM produktu WebSphere, na przykład na usługę LDAP.

Menedżery wdrażania

Menedżer wdrażania jest serwerem zarządzającym operacjami dla grupy logicznej lub komórki innych serwerów. Menedżer wdrażania stanowi centrum administrowania serwerami i klastrami.

Profil menedżera wdrażania jest pierwszym profilem, który należy utworzyć podczas tworzenia środowiska wdrażania. Każde utworzone środowisko wdrażania udostępnia konsolę Szybki start, za pomocą której można uruchamiać i zatrzymywać menedżer wdrażania, a także uruchamiać jego Konsolę administracyjną. Konsola administracyjna menedżera wdrażania służy do zarządzania serwerami i klastrami w komórce. Obejmuje to konfigurowanie serwerów i klastrów, dodawanie serwerów do klastrów, uruchamianie i zatrzymywanie serwerów i klastrów oraz wdrażanie modułów SCA.

Choć menedżer wdrażania jest pewnego rodzaju serwerem, nie można instalować modułów bezpośrednio w menedżerze wdrażania.

Węzły

Węzeł jest logiczną grupą serwerów zarządzanych.

Węzeł zazwyczaj odpowiada logicznemu lub fizycznemu systemowi komputerowemu o odrębnym adresie IP hosta. Węzły nie mogą obejmować wielu komputerów. Nazwa węzła jest zwykle taka sama, jak nazwa hosta komputera.

W topologii wdrożenia sieciowego mogą istnieć węzły zarządzane i niezarządzane. W węźle zarządzanym istnieje proces agenta węzła, który zarządza konfiguracją i serwerami węzła. W węzłach niezarządzanych nie ma agenta węzła.

Węzły zarządzane:

Węzeł zarządzany jest węzłem, który został stowarzyszony w ramach menedżera wdrażania, i który zawiera agent węzła. Może on również zawierać serwery zarządzane. W węzle zarządzanym można konfigurować i uruchamiać serwery zarządzane.

Serwery skonfigurowane w węzle zarządzanym tworzą zasoby środowiska wdrażania. Te serwery są tworzone, konfigurowane, uruchamiane, zatrzymywane, zarządzane i usuwane za pomocą Konsoli administracyjnej menedżera wdrażania.

Węzeł zarządzany zawiera agent węzła, który zarządza wszystkimi serwerami znajdującymi się w węzle.

Jeśli węzeł jest stowarzyszony, proces agenta węzła jest tworzony automatycznie. Aby można było zarządzać konfiguracją profilu, ten agent węzła musi być uruchomiony. Na przykład podczas wykonywania następujących czynności:

- Uruchamianie i zatrzymywanie procesów serwera.
- Synchronizowanie danych konfiguracji w menedżerze wdrażania z kopią w węzle.

Jednak do działania aplikacji lub konfigurowania zasobów w węzle nie jest konieczne uruchomienie agenta węzła.

Węzeł zarządzany może zawierać jeden lub więcej serwerów, zarządzanych za pomocą menedżera wdrażania. Na serwerach w węzle zarządzanym można wdrażać rozwiązania, ale węzeł zarządzany nie zawiera galerii przykładowych aplikacji. Węzeł zarządzany jest definiowany przez profil węzła zarządzanego i udostępnia konsolę Szybki start.

Węzły niezarządzane:

Węzeł niezarządzany nie ma agenta węzła, który zarządza jego serwerami.

Węzły niezarządzane w topologii wdrożenia sieciowego mogą mieć definicje serwerów takich jak serwery WWW, ale nie mogą mieć definicji serwerów aplikacji. Węzły niezarządzane nie mogą być stowarzyszone. Oznacza to, że do węzła niezarządzanego nie można dodać agenta węzła. Innym typem węzła niezarządzanego jest serwer autonomiczny. Menedżer wdrażania nie może zarządzać takim serwerem autonomicznym, ponieważ nie jest on znany w komórce. Serwer autonomiczny może być stowarzyszony. Kiedy jest stowarzyszony, tworzony jest automatycznie agent węzła. Węzeł staje się węzłem zarządzanym w komórce.

Agenty węzłów

Agenty węzłów są agentami administracyjnymi, które kierują żądania administracyjne do serwerów.

Agent węzła jest serwerem, który działa na każdym systemie hosta uczestniczącym w konfiguracji wdrożenia sieciowego. Jest to agent w pełni administracyjny i nie bierze udziału w funkcjach udostępniania aplikacji. Agent węzła udostępnia również inne ważne funkcje administracyjne, takie jak usługi przesyłania plików, synchronizację konfiguracji oraz monitorowanie wydajności.

Uwagi na temat nazewnictwa profili, węzłów, serwerów, hostów i komórek

W niniejszym temacie omówiono zastrzeżone terminy oraz kwestie, które należy uwzględnić podczas nadawania nazwy profilu, węzła, serwera, hosta i komórki (jeśli mają zastosowanie). Ten temat odnosi się do platform rozproszonych.

Uwagi na temat nazewnictwa profilu

Nazwa profilu może być dowolną unikalną nazwą z następującymi ograniczeniami. Podczas tworzenia nazwy profilu nie należy używać żadnych z następujących znaków:

- spacje,
- znaki specjalne, które nie są dozwolone w nazwach katalogów w systemie operacyjnym użytkownika, takie jak *, & lub ?,

- ukośniki (/) lub ukośniki odwrotne (\).

Znaki dwubajtowe są dozwolone.

Windows **Uwagi na temat ścieżki do katalogu:** Ścieżka do katalogu musi być mniejsza lub równa 60 znaków. Liczba znaków w katalogu `ścieżka_do_katalogu_profil\nazwa_profilu` musi być mniejsza lub równa 80 znaków.

Uwaga: Tworząc profil w środowisku Windows, należy stosować krótkie nazwy ścieżek, aby uniknąć przekroczenia limitu 255 znaków w ścieżce, który obowiązuje w systemie Windows.

Uwagi na temat nazewnictwa węzłów, serwerów, hostów i komórek

Nazwy zastrzeżone: należy unikać używania nazw zastrzeżonych jako wartości pól. Używanie nazw zastrzeżonych może prowadzić do nieprzewidywalnych wyników. Następujące słowa są zastrzeżone:

- komórki,
- węzły,
- serwery,
- klastry,
- aplikacje,
- wdrożenia.

Opisy pól na stronach Nazwy węzłów i hostów oraz Nazwy węzłów, hostów i komórek: podczas tworzenia profili należy stosować się do odpowiednich wytycznych dotyczących nazewnictwa.

- Profile serwera autonomicznego
- Profile menedżera wdrażania
- Profile węzła zarządzanego

Tabela 20. Wytyczne dotyczące nazewnictwa dla profili serwera autonomicznego

Nazwa pola	Wartość domyślna	Ograniczenia	Opis
Nazwa węzła	<p>Linux UNIX</p> <p>Windows <code>nazwaSkróconaHosta</code> <i>Węzeł NumerWęzła</i> gdzie:</p> <ul style="list-style-type: none"> • <i>nazwaSkróconaHosta</i> jest nazwą skróconą hosta. • <i>NumerWęzła</i> jest numerem kolejnym rozpoczynającym się od wartości 01. 	Należy unikać używania nazw zastrzeżonych.	Należy wybrać dowolną nazwę. Jeśli planowane jest utworzenie więcej niż jednego serwera w systemie, pomocne w uporządkowaniu instalacji będzie użycie nazwy unikalnej.
Nazwa serwera	<p>Linux UNIX</p> <p>Windows <code>serwer1</code></p>	Dla serwera należy użyć nazwy unikalnej.	Nazwa logiczna serwera.
Nazwa hosta	<p>Linux UNIX</p> <p>Windows Długa postać nazwy serwera nazw domen (DNS).</p>	Należy użyć pełnej nazwy hosta, która jest dostępna w używanej sieci.	Aby umożliwić komunikowanie się ze stacją roboczą, należy użyć jej bieżącej nazwy DNS lub adresu IP. Informacje dodatkowe o nazwie hosta są zawarte poniżej tabeli.

Tabela 21. Wytyczne dotyczące nazewnictwa dla profili menedżera wdrażania

Nazwa pola	Wartość domyślna	Ograniczenia	Opis
Nazwa węzła	<p>Linux UNIX</p> <p>Windows <i>nazwaSkróconaHosta</i></p> <p>Menedżer komórki <i>NumerWęzła</i> gdzie:</p> <ul style="list-style-type: none"> <i>nazwaSkróconaHosta</i> jest nazwą skróconą hosta. <i>NumerWęzła</i> jest numerem kolejnym rozpoczynającym się od wartości 01. 	Należy użyć unikalnej nazwy menedżera wdrażania. Należy unikać używania nazw zastrzeżonych.	Nazwa używana na potrzeby administrowania w ramach komórki menedżera wdrażania.
Nazwa hosta	<p>Linux UNIX</p> <p>Windows Długa postać nazwy serwera nazw domen (DNS).</p>	Należy użyć pełnej nazwy hosta, która jest dostępna w używanej sieci. Należy unikać używania nazw zastrzeżonych.	Aby umożliwić komunikowanie się ze stacją roboczą, należy użyć jej bieżącej nazwy DNS lub adresu IP. Informacje dodatkowe o nazwie hosta są zawarte poniżej tabeli.
Nazwa komórki	<p>Linux UNIX</p> <p>Windows <i>nazwaSkróconaHosta</i></p> <p>Komórka <i>NumerKomórki</i> gdzie:</p> <ul style="list-style-type: none"> <i>nazwaSkróconaHosta</i> jest nazwą skróconą hosta. <i>NumerKomórki</i> jest numerem kolejnym rozpoczynającym się od wartości 01. 	Należy użyć unikalnej nazwy komórki menedżera wdrażania. Nazwa komórki musi być unikalna w każdym przypadku, w którym produkt działa na tej samej fizycznej stacji roboczej lub w tym samym klastrze składającym się ze stacji, na przykład Sysplex. Ponadto nazwa komórki musi być unikalna w każdej sytuacji, w której wymagane jest połączenie sieciowe między jednostkami (między komórkami lub od klienta, który musi się komunikować ze wszystkimi komórkami). Nazwy komórek również muszą być unikalne, jeśli ich przestrzenie nazw mają zostać stowarzyszone. W przeciwnym razie mogą wystąpić takie objawy, jak wyjątek <code>javax.naming.NameNotFoundException</code> . W tym przypadku należy utworzyć komórki o unikalnych nazwach.	Wszystkie węzły stowarzyszone stają się elementami komórki menedżera wdrażania nazwanej na stronie Nazwy węzłów, hostów i komórek narzędzia Profile Management Tool.

Tabela 22. Wytyczne dotyczące nazewnictwa profili węzła zarządzanego

Nazwa pola	Wartość domyślna	Ograniczenia	Opis
Nazwa węzła	<div style="display: flex; justify-content: space-around;"> Linux UNIX </div> <div style="display: flex; justify-content: space-around;"> Windows </div> <p><i>nazwaSkróconaHosta</i> Węzeł <i>NumberWęzła</i> gdzie:</p> <ul style="list-style-type: none"> <i>nazwaSkróconaHosta</i> jest nazwą skróconą hosta. <i>NumerWęzła</i> jest numerem kolejnym rozpoczynającym się od wartości 01. 	<p>Należy unikać używania nazw zastrzeżonych.</p> <p>W obrębie komórki menedżera wdrażania należy używać nazwy unikalnej.</p>	<p>Nazwa jest używana do administrowania w komórce menedżera wdrażania, do której dodano profil węzła zarządzanego. W obrębie komórki menedżera wdrażania należy używać nazwy unikalnej.</p>
Nazwa hosta	<div style="display: flex; justify-content: space-around;"> Linux UNIX </div> <div style="display: flex; justify-content: space-around;"> Windows </div> <p>Długa postać nazwy serwera nazw domen (DNS).</p>	<p>Należy użyć pełnej nazwy hosta, która jest dostępna w używanej sieci.</p>	<p>Aby umożliwić komunikowanie się ze stacją roboczą, należy użyć jej bieżącej nazwy DNS lub adresu IP. Informacje dodatkowe o nazwie hosta są zawarte poniżej tabeli.</p>

Uwagi na temat nazwy hosta:

Nazwa hosta jest nazwą sieciową stacji fizycznej, na której jest zainstalowany węzeł. Nazwa hosta musi być tłumaczona na fizyczny węzeł sieci na serwerze. Jeśli serwer na wiele kart sieciowych, nazwa hosta lub adres IP muszą być tłumaczone na jedną z kart sieciowych. Nazwa hosta jest używana przez węzły zdalne do nawiązywania połączenia i komunikowania się z danym węzłem.

IBM Business Process Manager jest zgodny z protokołem IPv4 (Internet Protocol, wersja 4) i IPv6 (Internet Protocol, wersja 6). Wprowadzanie adresów IP w Konsoli administracyjnej lub w innym miejscu można odbywać się w dowolnym z wymienionych formatów. Należy zauważyć, że jeśli w systemie został zaimplementowany protokół IPv6, adres IP należy wprowadzić w formacie IPv6 i na odwrót, jeśli protokół IPv6 nie jest jeszcze dostępny, należy wprowadzać adresy IP w formacie IPv4. Więcej informacji na temat protokołu IPv6 zawiera następujący opis: IPv6.

Następujące wytyczne mogą pomóc w określeniu odpowiedniej nazwy hosta dla stacji roboczej:

- Wybierz nazwę hosta, która jest dostępna dla innych stacji roboczych znajdujących się w sieci.
- Jako nazwy hosta nie używaj identyfikatora ogólnego (localhost).
- Nie próbuj instalować produktów IBM Business Process Manager na serwerze o nazwie hosta zawierającej znaki z zestawu znaków dwubajtowych (DBCS). Znaki DBCS nie są obsługiwane dla nazw hostów.
- W nazwach serwerów unikaj używania znaków podkreślenia (_). Standardy internetowe wymuszają, aby używane nazwy domenowe były zgodne z wymaganiami obowiązującymi dla nazw hostów. Wymagania te zostały opisane w dokumentach Internet Official Protocol Standards RFC 952 i RFC 1123. Nazwy domenowe mogą zawierać tylko litery (wielkie lub małe) oraz cyfry. Nazwy domenowe mogą także zawierać myślniki (-), jeśli nie znajdują się one na końcu nazwy. Znaki podkreślenia (_) w nazwie hosta nie są obsługiwane. Jeśli produkt IBM Business Process Manager został zainstalowany na serwerze, którego nazwa zawiera znak podkreślenia, do czasu zmiany nazwy dostęp do serwera należy uzyskiwać za pomocą jego adresu IP.

Jeśli na tym samym komputerze definiowane są współistniejące węzły o unikanych adresach IP, każdy adres IP należy zdefiniować w tabeli wyszukiwania serwera nazw domen (DNS). Pliki konfiguracyjne dla serwerów nie udostępniają tłumaczenia nazw domenowych dla wielu adresów IP na stacji roboczej z jednym adresem sieciowym.

Wartość określona dla nazwy hosta jest używana jako wartość właściwości `hostname` w dokumentach związanych z konfiguracją. Wartość nazwy hosta należy określić w jednym z następujących formatów:

- łańcuch pełnej nazwy hosta serwerów nazw domen (DNS), na przykład `xmachine.manhattan.ibm.com`,
- łańcuch domyślnej skróconej nazwy hosta DNS, na przykład `xmachine`,
- liczbowy adres IP, na przykład `127.1.255.3`.

Pełna nazwa hosta DNS ma tę zaletę, że jest całkowicie jednoznaczna i elastyczna. Istnieje możliwość zmiany aktualnego adresu IP systemu hosta bez konieczności zmieniania konfiguracji serwera. Określenie wartości dla nazwy hosta jest szczególnie użyteczne, jeśli planowana jest częsta zmiana adresu IP w sytuacji, gdy do przydzielania adresów IP jest używany protokół DHCP (Dynamic Host Configuration Protocol). Wadą tego formatu jest uzależnienie od systemu DNS. Jeśli system DNS nie jest dostępny, połączenie zostanie zerwane.

Skrócona nazwa hosta jest również tłumaczona dynamicznie. Dla formatu nazwy skróconej istnieje dodatkowa możliwość jego ponownego zdefiniowania w lokalnym pliku hosts, więc system może uruchomić serwer nawet wtedy, gdy jest on odłączony od sieci. Aby można było uruchomić serwer po odłączeniu od sieci, nazwę skróconą należy zdefiniować na wartość 127.0.0.1 (lokalna pętla zwrotna) w pliku hosts. Wadą formatu nazwy skróconej jest uzależnienie od systemu DNS w przypadku dostępu zdalnego. Jeśli system DNS nie jest dostępny, połączenie zostanie zerwane.

Liczbowy adres IP ma tę zaletę, że nie wymaga tłumaczenia nazw przez system DNS. Węzeł zdalny może nawiązać połączenie z węzłem nazwanym liczbowym adresem IP, gdy system DNS nie jest dostępny. Wadą tego formatu jest to, że liczbowy adres IP jest stały. Po każdej zmianie adresu IP stacji roboczej należy zmienić ustawienie właściwości hostName w dokumentach konfiguracyjnych. Dlatego też w przypadku używania DHCP lub częstej zmiany adresów IP nie należy korzystać z liczbowego adresu IP. Inną wadą tego formatu jest brak możliwości użycia węzła, jeśli host jest rozłączony z siecią.

Specyfikacja BPMN 2.0

Definicje procesów biznesowych w produkcie IBM Business Process Manager obsługują podklasę Common Executable klasy zgodności Process Modeling specyfikacji BPMN 2.0, która jest związana z modelami, które można uruchomić.

Specyfikacja BPMN (Business Process Model and Notation) jest fundamentalnym standardem dotyczącym procesów w produktach IBM Process Designer oraz IBM Process Center. Diagramy definicji procesów biznesowych (Business Process Definition - BPD) są oparte na specyfikacji BPMN. W tym temacie opisano niektóre sposoby zastosowania specyfikacji BPMN 2.0 w produkcie IBM Business Process Manager. Więcej informacji na temat specyfikacji BPMN można znaleźć na poświęconej jej stronie <http://www.bpmn.org/>.

IBM Business Process Manager obsługuje następujące typy czynności specyfikacji BPMN 2.0:

- Brak (None - czynność abstrakcyjna w specyfikacji BPMN 2.0)
- Zadanie systemowe (System Task - Service Task w specyfikacji BPMN 2.0)
- Czynność użytkownika (User Task)
- Skrypt (Script)
- Zadanie decyzyjne (Decision Task - Business Rule Task w specyfikacji BPMN 2.0)

Zdarzenia pośrednie komunikatu w produkcie IBM BPM udostępniają funkcje podobne do czynności Send Task i Receive Task w specyfikacji BPMN.

Notacja BPMN 2.0

Począwszy od wersji 7.5.1 ikony czynności na diagramach BPD w produkcie Process Designer zgodne ze specyfikacją BPMN 2.0 zostały zebrane na uproszczonej palecie i są wyświetlane w diagramach procesów. Dzięki tym ikonom można stwierdzić, czy dane działanie jest działaniem systemowym, czynnością użytkownika, zadaniem decyzyjnym, skryptem, czy procesem połączonym. Działania znajdujące się w modelach utworzonych z użyciem wcześniejszych wersji produktu także są reprezentowane za pomocą odpowiednich typów i ikon czynności zgodnych ze specyfikacją BPMN 2.0, gdy są wyświetlane w wersji 7.5.1 lub nowszej.

Działania i czynności

W stosunku do poprzednich wersji produktu Process Designer wprowadzono pewne zmiany terminologiczne. Niektóre z nich dotyczą zmiany nazw typów działań.

- Działania usługi (zautomatyzowane) są obecnie zadaniami systemowymi.
- Działania usługi (zadania) na torze innym niż systemowy są obecnie czynnościami użytkownika.
- Działania usługi (zadania) na torze systemowym są obecnie zadaniami decyzyjnymi, jeśli odwołują się do usługi decyzyjnej.
- Działania usługi (zadania) na torze systemowym są obecnie zadaniami systemowymi, jeśli odwołują się do usługi dowolnego rodzaju innego niż usługa decyzyjna.
- Działania Javascript są obecnie działaniami skryptu.
- Zagnieżdżone działania procesu są obecnie procesami połączonymi.
- Działania zewnętrzne z poprzednich wersji produktu Process Designer są dostępne jako zewnętrzne implementacje czynności użytkownika i zadań systemowych.

Bramki

W stosunku do poprzednich wersji nie nastąpiły żadne zmiany notacji dotyczące bramek. Występują jednak trzy zmiany terminologiczne. Bramka decyzyjna jest obecnie *bramką wykluczającą*, prosta bramka podziału lub łączenia jest obecnie *bramką równoległą*, a warunkowa bramka podziału lub łączenia jest obecnie *bramką obejmującą*.

Pojawił się także nowy typ bramki: *bramka zdarzeń*. Bramka zdarzeń reprezentuje punkt rozgałęzienia w procesie, w którym wychodzące z niego alternatywne ścieżki są wybierane na podstawie występującego zdarzenia, a nie w wyniku wartościowania wyrażen z użyciem danych procesu (jak ma to miejsce w przypadku bramek wykluczających i obejmujących). Wybierana ścieżka jest określana na podstawie konkretnego zdarzenia, zwykle odebrania komunikatu.

Zdarzenia nieprzerywające

W specyfikacji BPMN 2.0 dodano notację na potrzeby zdarzeń nieprzerywających. Domyślnie zdarzenie na granicy przerywa działanie, do którego jest przyłączone. W momencie wyzwolenia zdarzenia działanie zostaje zatrzymane, a znacznik przechodzi dalej w kierunku wychodzącego przepływu sekwencji związanej ze zdarzeniem. Gdy zdarzenie jest ustawione jako nieprzerywające, po wyzwoleniu zdarzenia wykonywanie przyłączonego działania jest kontynuowane równoległe, a nowy znacznik zostaje wygenerowany i przekazany dalej w kierunku wychodzącego przepływu sekwencji związanej ze zdarzeniem. Granicą zdarzenia w przypadku zdarzeń nieprzerywających jest linia kreskowana.

Zdarzenia pośrednie przyłączone do działań są przerywającymi zdarzeniami pośrednimi, jeśli zamykają przyłączone do nich działania, lub nieprzerywającymi zdarzeniami pośrednimi, jeśli nie zamykają przyłączonych do nich działań.

Zdarzenie uruchomienia

Specyfikacja BPMN pozwala na pominięcie w modelach procesów symboli zdarzeń uruchomienia i zakończenia. W produkcie Process Designer wymagana jest obecność zdarzeń uruchomienia i zakończenia w modelach procesów.

W produkcie Process Designer dostępne są różne typy zdarzeń uruchomienia:

procesy

- brak
- komunikat
- doraźne

podprocesy

- brak

podprocesy zdarzenia

- błąd
- komunikat
- licznik czasu

Istnieje możliwość zmiany typu zdarzenia uruchomienia przez edycję właściwości zdarzenia. W procesie może występować wiele zdarzeń uruchomienia przez komunikat, ale można używać tylko jednego ogólnego zdarzenia uruchomienia.

Zdarzenia zakończenia

Dostępne są cztery typy zdarzenia zakończenia: *komunikat*, *przerwanie*, *błąd* i *brak*. Typ zdarzenia zakończenia można zmienić.

Kiedy proces nadrzędny wywołuje proces potomny, a proces potomny uruchamia działanie związane ze zdarzeniem przerwania, proces potomny zatrzymuje się, a proces nadrzędny kontynuuje działanie w kolejnych krokach.

Podprocesy

Specyfikacja BPMN definiuje dwa typy podprocesów: osadzone i wielokrotnego użytku. Oba rodzaje można tworzyć w produkcie Process Designer. Procesy osadzone są nazywane po prostu *podprocesami* w produkcie Process Designer, a w wersji 7.5.1 są nowymi obiektami. Podproces wielokrotnego użytku w specyfikacji BPMN jest nazywany *procesem połączonym* w produkcie Process Designer.

Podproces istnieje w obrębie procesu, który go zawiera, i pozwala na grupowanie kroków procesu w celu zmniejszenia złożoności i uporządkowanie diagramu. Za pomocą podprocesów można w jednym działaniu zwinąć wiele kroków. Podproces jest widoczny jedynie dla procesu, w którym jest zdefiniowany. Podproces istnieje w zasięgu procesu wywołującego i ma dostęp do wszystkich zmiennych w obrębie środowiska. Nie stosuje się mechanizmu przekazywania parametrów do i z procesu osadzonego.

Oprócz podprocesów i procesów połączonych w produkcie Process Designer dostępne są podprocesy zdarzeń. Podproces zdarzenia jest to specjalizowany proces przeznaczony do obsługi zdarzeń. Nie jest on połączony z innymi działaniami za pośrednictwem przepływu sekwencji i występuje tylko, jeśli zostanie wyzwolone jego zdarzenie uruchomienia.

Procesy połączone

Podproces wielokrotnego użytku w specyfikacji BPMN jest nazywany *procesem połączonym* w produkcie Process Designer. Jest to proces utworzony poza bieżącym procesem, który może być wywoływany przez bieżący proces. Jest to proces wielokrotnego użytku, ponieważ może być wywoływany także przez inne definicje procesów. Proces połączony definiuje swoje parametry wejściowe oraz wyjściowe i nie ma dostępu do zasięgu ani środowiska procesu wywołującego. Proces połączony jest podobny do procesu zagnieżdżonego dostępnego we wcześniejszych wersjach. Jego działanie nie uległo zmianie. Wcześniejsze procesy zagnieżdżone zostaną poddane migracji do postaci procesów połączonych. Proces połączony wygląda jak podproces o grubym zarysie, który jest podświetlony w oknie komponentu Inspector.

Pętle

Specyfikacja BPMN udostępnia pojęcie działania, które można powtarzać. Działanie może być niepodzielne, co oznacza, że jest powtarzane, lub może być podprocesem obudowującym ciąg powtarzanych kroków. Po rozwinięciu działania powtarzanego widoczne stają się zawarte w nim działania, których uruchamianie jest powtarzane. Warunek jest zawsze wartościowany na początku każdej iteracji pętli. Nie ma możliwości wartościowania na końcu każdej iteracji pętli.

IBM Business Process Manager udostępnia *pętlę obejmującą wiele instancji* uruchamianą skończoną liczbą razy wraz z zawartymi w niej działaniami, które są uruchamiane sekwencyjnie lub równoległe.

Importowanie procesów innych niż procesy BPMN

Istnieje możliwość zaimportowania modeli, które utworzono w produkcie IBM WebSphere Business Modeler, i używania ich w produkcie Process Designer. Informacje na temat importu BPMN 2.0 zawiera sekcja *Odwzorowywanie*

elementów produktu IBM WebSphere Business Modeler w konstrukcje produktu IBM Business Process Manager. Można także importować modele zgodne ze specyfikacją BPMN 2.0 utworzone w produktach IBM WebSphere Business Compass, Rational Software Architect lub w innych środowiskach modelowania.

Definicje procesów biznesowych (BPD)

Aby utworzyć model procesu w produkcie IBM Process Designer, należy utworzyć definicję procesu biznesowego (BPD). Definicja procesu biznesowego może być oparta na zaimportowanym modelu BPMN.

Definicja BPD jest modelem procesu wielokrotnego użytku określającym wspólne elementy wszystkich instancji danego modelu procesu w środowisku wykonawczym. Definicja BPD musi obejmować zdarzenie uruchomienia, zdarzenie zakończenia, przynajmniej jeden tor i jedno lub więcej działań. Szczegółowe informacje na temat ograniczeń dotyczących znaków, które obowiązują w definicjach BPD, zawiera sekcja „Konwencje nazewnictwa stosowane w produkcie IBM Process Designer”.

W definicji procesu biznesowego (Business Process Definition - BPD) potrzebny jest tor dla każdego systemu lub każdej grupy użytkowników, uczestniczących w procesie. Tor może być torem uczestnika lub torem systemowym. Można jednak utworzyć definicję BPD, w której działania grupy i systemu są umieszczone na pojedynczym torze, jeśli takie są preferencje użytkownika. Informacje dotyczące tworzenia definicji BPD można znaleźć w temacie „Tworzenie definicji procesu biznesowego (BPD)” dostępnym w sekcji tematów pokrewnych.

Można wyznaczyć dowolną osobę lub grupę, która będzie odpowiedzialna za działania znajdujące się na torze uczestnika. Każdy tworzony tor jest domyślnie przypisywany do grupy Wszyscy użytkownicy. Z tej domyślnej grupy można korzystać przy uruchamianiu i testowaniu definicji BPD w komponencie Inspector. Grupa Wszyscy użytkownicy obejmuje wszystkich użytkowników należących do grupy zabezpieczeń **tw_allusers** (jest to specjalna grupa zabezpieczeń obejmująca automatycznie wszystkich użytkowników systemu).

Tor systemowy zawiera działania obsługiwane przez określony system komponentu IBM Process Center. Każde działanie wymaga implementacji, która definiuje dane działanie oraz ustawia właściwości czynności bądź zadania. W trakcie implementowania programista tworzy usługę lub pisze kod w języku JavaScript niezbędny do wykonania działań w torze systemowym. Informacje dotyczące usług można znaleźć w temacie „Podstawowe informacje o typach usług” dostępnym w sekcji tematów pokrewnych.

Dla każdej tworzonej definicji BPD należy zadeklarować zmienne służące do przechwytywania danych biznesowych przekazywanych między działaniami. Informacje dotyczące implementowania zmiennych można znaleźć w temacie „Zarządzanie zmiennymi i ich odwzorowywanie” dostępnym w sekcji tematów pokrewnych.

Do definicji BPD można dodawać także zdarzenia. Zdarzenia w produkcie IBM BPM mogą być wyzwalane w wyniku minięcia określonej daty zakończenia, w odpowiedzi na wyjątek lub jako reakcja na nadchodzący komunikat. Żądany wyzwalacz determinuje typ zdarzenia, które należy zaimplementować. Szczegółowe informacje na temat dostępnych typów zdarzeń i ich wyzwalaczy zawiera sekcja „Modelowanie zdarzeń”.

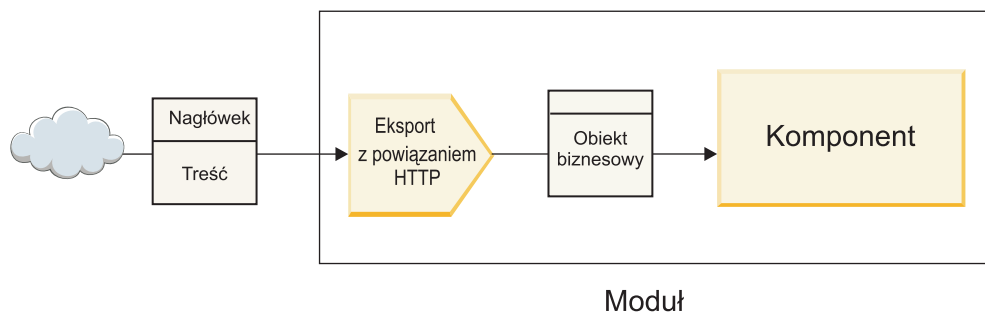
Powiązania

Podstawą architektury zorientowanej na usługi jest pojęcie *usługi* - jednostki funkcjonalności realizowanej poprzez interakcję między urządzeniami komputerowymi. *Eksport* definiuje zewnętrzny interfejs (lub punkt dostępu) modułu, tak aby komponenty architektury SCA (Service Component Architecture) w module mogły udostępniać swoje usługi klientom zewnętrznym. *Import* definiuje interfejs z usługami znajdującymi się poza modulem, aby usługi mogły być wywoływane z wewnątrz modułu. *Powiązania* specyficzne dla protokołu są używane z importami i eksportami w celu określenia sposobu transportowania danych do modułu lub z modułu.

Eksperty

Klienty zewnętrzne mogą wywoływać komponenty SCA w module integracji przy użyciu różnych protokołów (na przykład HTTP, JMS, MQ lub RMI/IIOP) oraz danych w różnych formatach (na przykład XML, CSV, COBOL lub JavaBeans). Eksperty są komponentami, które odbierają te żądania ze źródeł zewnętrznych, a następnie wywołują komponenty produktu IBM Business Process Manager przy użyciu modelu programistycznego SCA.

Na poniższej ilustracji przedstawiono przykład, w którym eksport odbiera żądanie z aplikacji klienckiej za pośrednictwem protokołu HTTP. Dane są transformowane w obiekt biznesowy - format używany przez komponent SCA. Komponent jest następnie wywoływany przy użyciu tego obiektu danych.

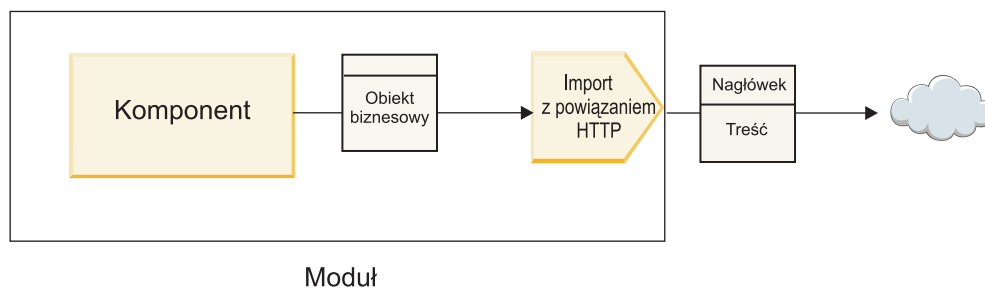


Rysunek 1. Eksport z powiązaniem HTTP

Importy

Użytkownik może wymagać, aby komponent SCA wywoływał usługę zewnętrzną inną niż SCA, która oczekuje danych w innym formacie. Komponent SCA używa importu, aby wywołać usługę zewnętrzną przy użyciu modelu programistycznego SCA. Import następnie wywołuje docelową usługę w sposób oczekiwany przez tę usługę.

Na poniższej ilustracji przedstawiono przykład, w którym żądanie jest wysyłane z komponentu SCA za pośrednictwem importu do usługi zewnętrznej. Obiekt biznesowy (format używany przez komponent SCA) jest transformowany w format oczekiwany przez usługę, a następnie wywoływana jest ta usługa.



Rysunek 2. Import z powiązaniem HTTP

Lista powiązań

Do generowania powiązania dla importu lub eksportu oraz do konfigurowania powiązania używany jest produkt Integration Designer. Poniższa lista opisuje typy dostępnych powiązań.

- SCA

Powiązanie SCA (domyślne) umożliwia usłudze komunikację z usługami w innych modułach SCA. Aby uzyskać dostęp do usługi w innym module SCA, należy użyć importu z powiązaniem SCA. Aby zaoferować usługę innym modułom SCA, należy użyć eksportu z powiązaniem SCA.
- Usługa Web Service

Powiązanie usługi Web Service pozwala uzyskać dostęp do usługi zewnętrznej przy użyciu współdziałających komunikatów SOAP i jakości usług. Powiązań usługi Web Service można także użyć, aby dołączyć załączniki do komunikatu SOAP.

Powiązanie usługi Web Service może używać protokołu transportowego SOAP/HTTP (SOAP przez HTTP) lub SOAP/JMS (SOAP przez JMS). Niezależnie od transportu używanego do przekazania komunikatów SOAP (HTTP lub JMS) powiązania usług Web Services zawsze obsługują interakcję żądanie-odpowiedź w sposób synchroniczny.
- HTTP

Powiązanie HTTP pozwala uzyskać dostęp do usługi zewnętrznej przy użyciu protokołu HTTP w sytuacjach, kiedy używane są komunikaty inne niż SOAP lub kiedy wymagany jest bezpośredni dostęp HTTP. To powiązanie jest używane, jeśli użytkownik pracuje z usługami Web Services opartymi na modelu HTTP (czyli usługami korzystającymi z powszechnie znanych operacji interfejsu HTTP takich jak GET, PUT, DELETE itp.).

- Komponenty EJB (Enterprise JavaBeans)

Powiązania komponentów EJB pozwalają komponentom SCA na interakcję z usługami udostępnianymi przez logikę biznesową Java EE działającą na serwerze Java EE.

- EIS

Powiązanie EIS (Enterprise Information System) używane z adapterem zasobu JCA umożliwia dostęp do usług w systemie EIS lub udostępnienie usług systemowi EIS.

- Powiązania JMS

Powiązania JMS (Java Message Service), ogólne powiązania JMS i powiązania MQ JMS (WebSphere MQ JMS) służą do interakcji z systemami przesyłania komunikatów, w których asynchroniczna komunikacja za pośrednictwem kolejek komunikatów jest niezbędna do zachowania niezawodności.

Eksport z jednym z powiązań JMS obserwuje kolejkę, oczekując nadejścia komunikatu i asynchronicznie wysyła odpowiedź do kolejki odpowiedzi (jeśli odpowiedź jest potrzebna). Import z jednym z powiązań JMS buduje i wysyła komunikat do kolejki JMS i obserwuje kolejkę, oczekując nadejścia odpowiedzi (jeśli odpowiedź jest potrzebna).

- JMS

- Powiązanie JMS umożliwia dostęp do wbudowanego dostawcy JMS produktu WebSphere.

- Ogólne JMS

- Ogólne powiązanie JMS umożliwia dostęp do systemu przesyłania komunikatów od dostawcy innego niż IBM.

- MQ JMS

- Powiązanie MQ JMS umożliwia dostęp do podzbioru JMS systemu przesyłania komunikatów WebSphere MQ. To powiązanie jest używane, jeśli podzbiór funkcji JMS jest wystarczający na potrzeby aplikacji.

- MQ

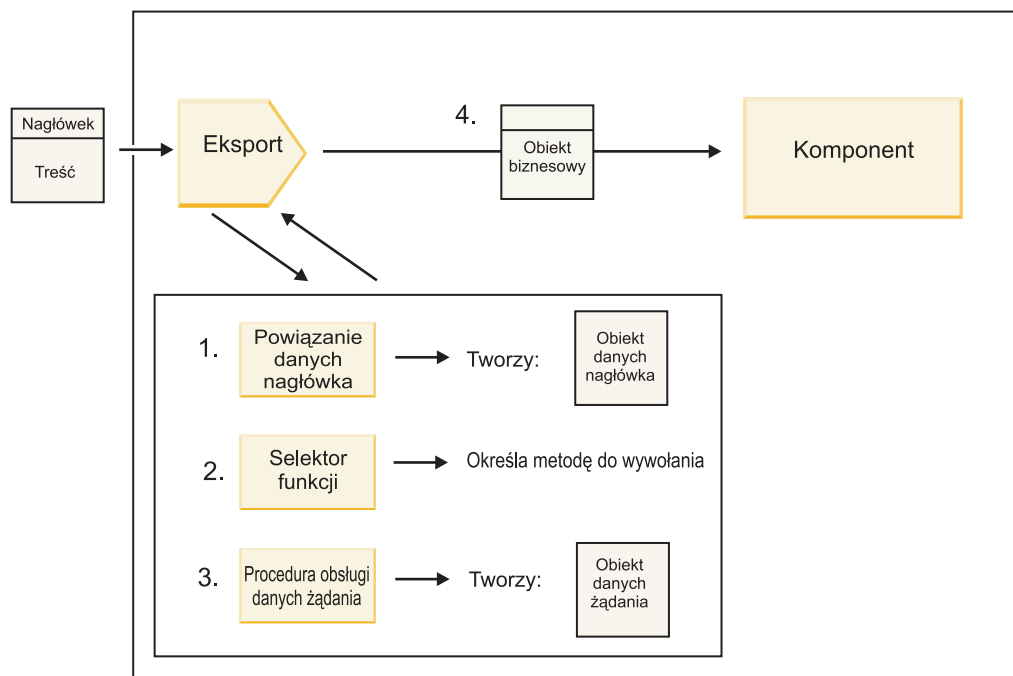
Powiązanie WebSphere MQ umożliwia komunikację z rodzimymi aplikacjami MQ, wprowadzając je do struktury architektury zorientowanej na usługi i zapewniając dostęp do informacji nagłówka specyficznych dla produktu MQ. To powiązanie jest używane, jeśli konieczne jest użycie rodzimych funkcji produktu MQ.

Przegląd powiązań eksportów i importów

Eksport umożliwia udostępnianie usług w module integracji klientom zewnętrznym. Import umożliwia komponentom SCA w module integracji wywoływanie usług zewnętrznych. Powiązanie związane z eksportem lub importem określa relację między komunikatami protokołu i obiektami biznesowymi. Określa również sposób wyboru operacji i błędów.

Przepływ informacji przez eksport

Eksport odbiera żądanie przeznaczone dla komponentu, z którym połączony jest ten eksport za pośrednictwem konkretnego transportu określanego przez związane z eksportem powiązanie (na przykład powiązanie HTTP).



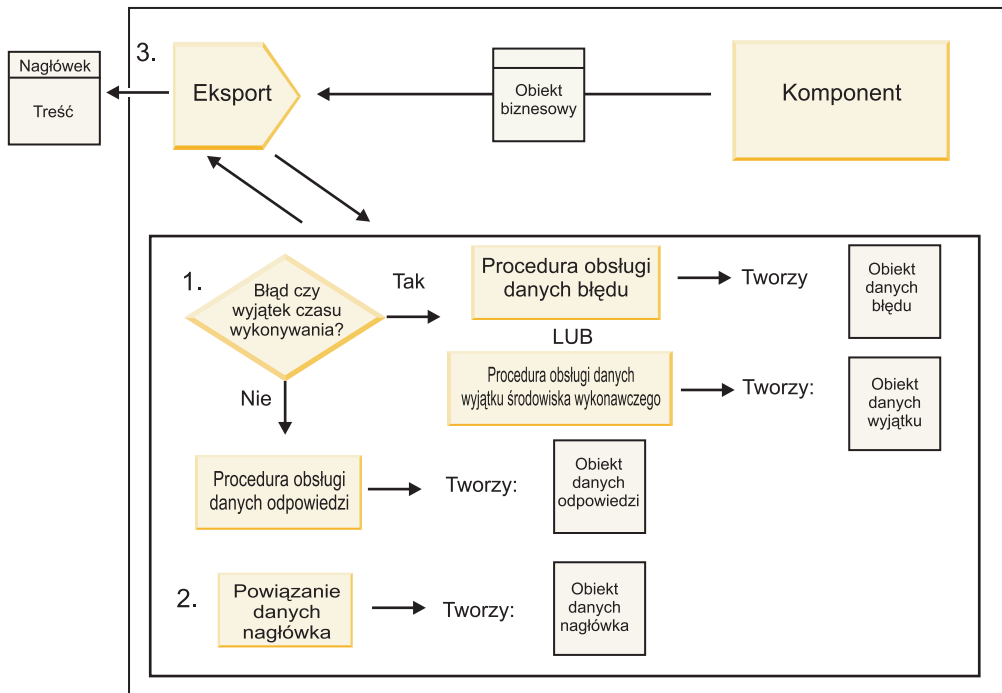
Rysunek 3. Przepływ żądania przez eksport do komponentu

Kiedy eksport odbierze żądanie, ma miejsce następująca sekwencja zdarzeń:

1. Tylko w przypadku powiązań WebSphere MQ: powiązanie danych nagłówka transformuje nagłówek protokołu w obiekt danych nagłówka.
2. Selektor funkcji określa nazwę metody rodzimej na podstawie komunikatu protokołu. Nazwa metody rodzimej jest odwzorowywana przez konfigurację eksportu na nazwę operacji interfejsu eksportu.
3. Procedura obsługi danych żądania lub powiązanie danych w metodzie transformują żądanie w obiekt biznesowy żądania.
4. Eksport wywołuje metodę komponentu z obiektem biznesowym żądania.
 - Powiązanie eksportu HTTP, powiązanie eksportu komponentu EJB i powiązanie eksportu usługi Web Service wywołują komponent SCA synchronicznie.
 - Powiązanie JMS, powiązanie ogólne JMS, powiązanie MQ JMS i powiązanie WebSphere MQ dla eksportu wywołują komponent SCA asynchronicznie.

Należy zauważyć, że eksport może propagować nagłówki i właściwości użytkownika otrzymane za pośrednictwem protokołu, jeśli włączone jest propagowanie kontekstu. Komponenty połączone z eksportem mogą następnie uzyskiwać dostęp do tych nagłówków i właściwości użytkownika. Więcej informacji zawiera temat “Propagacja” w Centrum informacyjnym produktu WebSphere Integration Developer.

Jeśli jest to operacja dwukierunkowa, komponent zwraca odpowiedź.



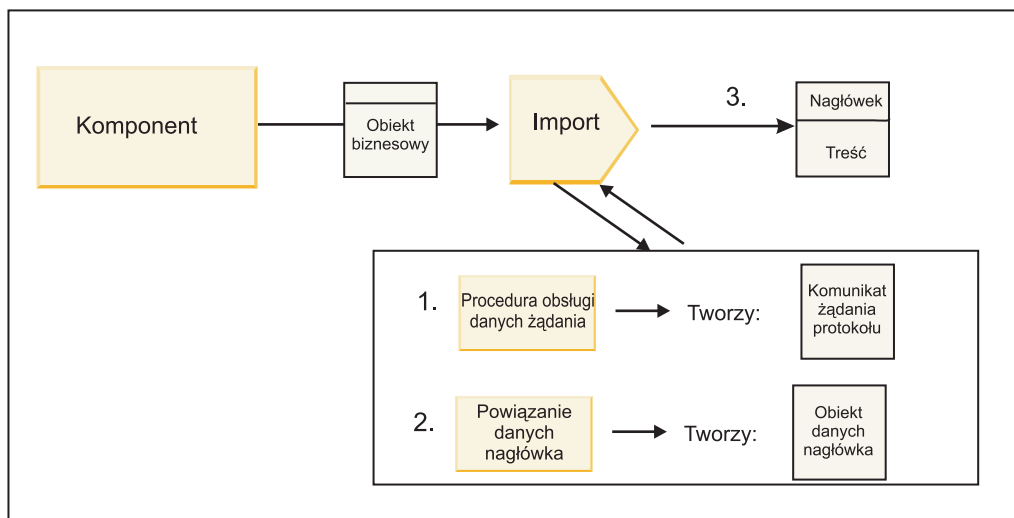
Rysunek 4. Przepływ odpowiedzi z powrotem przez eksport

W tym przypadku ma miejsce następująca sekwencja kroków:

1. Jeśli powiązanie eksportu otrzyma normalny komunikat odpowiedzi, procedura obsługi danych odpowiedzi lub powiązanie danych w metodzie transformują obiekt biznesowy w odpowiedź.
Jeśli odpowiedź jest błędem, procedura obsługi danych błędu lub powiązanie danych w metodzie transformują błąd w odpowiedź błędu.
Tylko w przypadku powiązań eksportu HTTP: jeśli odpowiedź jest wyjątkiem czasu wykonania, wywoływana jest procedura obsługi danych wyjątku czasu wykonania (jeśli jest skonfigurowana).
2. Tylko w przypadku powiązań WebSphere MQ: powiązanie danych nagłówka transformuje obiekty danych nagłówka w nagłówek protokołu.
3. Eksport wysyła odpowiedź usługi przez transport.

Przepływ informacji przez import

Komponenty wysyłają żądania do usług znajdujących się poza modulem za pośrednictwem importu. Żądanie jest wysyłane za pośrednictwem konkretnego transportu, określanego przez związane z importem powiązanie.



Rysunek 5. Przepływ z komponentu, przez import, do usługi

Komponent wywołuje import przy użyciu obiektu biznesowego żądania.

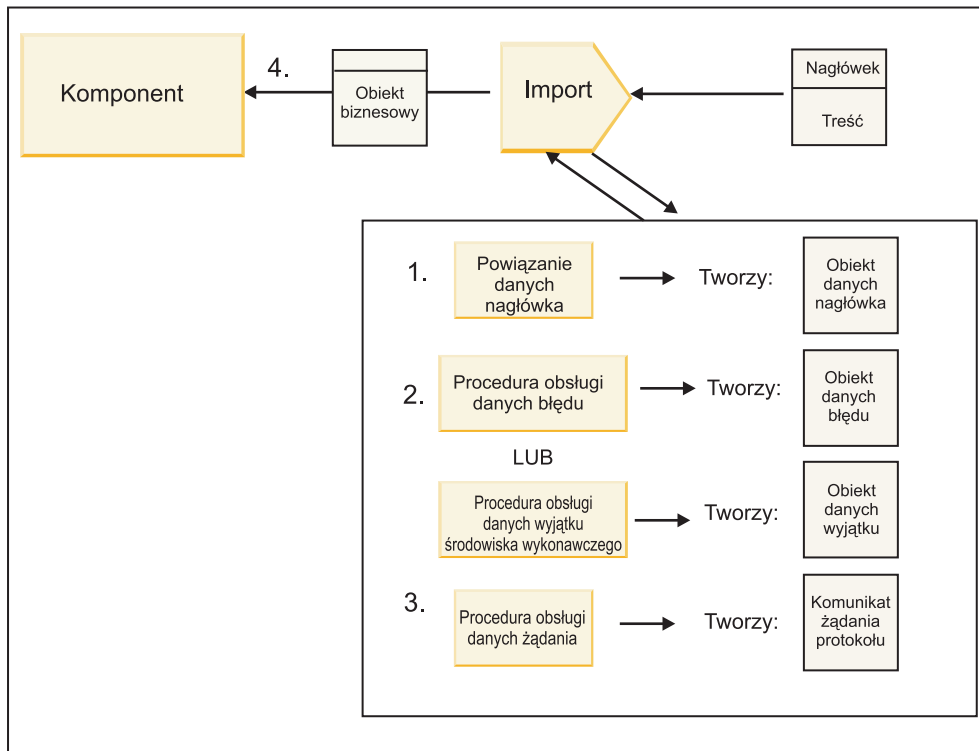
Uwaga:

- Powiązanie importu HTTP, powiązanie importu usługi Web Service i powiązanie importu komponentu EJB powinny być wywoływane synchronicznie przez komponent wywołujący.
- Powiązanie JMS, powiązanie ogólne JMS, powiązanie MQ JMS i powiązanie WebSphere MQ dla importu powinny być wywoływane asynchronicznie.

Po wywołaniu importu przez komponent ma miejsce następująca sekwencja zdarzeń:

1. Procedura obsługi danych żądania lub powiązanie danych w metodzie transformują obiekt biznesowy żądania w komunikat żądania protokołu.
2. Tylko w przypadku powiązań WebSphere MQ: powiązanie danych nagłówka w metodzie ustawia obiekt biznesowy nagłówka w nagłówku protokołu.
3. Import wywołuje usługę przy użyciu żądania usługi za pośrednictwem transportu.

Jeśli jest to operacja dwukierunkowa, usługa zwraca odpowiedź i ma miejsce następująca sekwencja kroków:



Rysunek 6. Przepływ odpowiedzi z powrotem przez import

1. Tylko w przypadku powiązań WebSphere MQ: powiązanie danych nagłówka transformuje nagłówek protokołu w obiekt danych nagłówka.
2. System określa, czy odpowiedź jest błędem.
 - Jeśli odpowiedź jest błędem, selektor błędu bada błąd, aby określić, na jaki błąd WSDL jest on odwzorowywany. Następnie procedura obsługi danych błędu w metodzie transformuje błąd w odpowiedź błędu.
 - Jeśli odpowiedź jest wyjątkiem czasu wykonania, wywoływana jest procedura obsługi danych wyjątku czasu wykonania (jeśli jest skonfigurowana).
3. Procedura obsługi danych odpowiedzi lub powiązanie w metodzie transformują odpowiedź w obiekt biznesowy odpowiedzi.
4. Import zwraca obiekt biznesowy odpowiedzi do komponentu.

Konfiguracja powiązania eksportu i importu

Jednym z najważniejszych aspektów powiązań eksportów i importów jest możliwość transformacji formatu danych, która określa sposób odwzorowania danych (deserializowania) z rodzimego formatu łącznika na obiekt biznesowy lub sposób odwzorowania danych (serializowania) z obiektu biznesowego na rodzimy format łącznika. W przypadku powiązań związanych z eksportami można również określić selektor funkcji, który określa operację, jaka powinna być wykonana na danych. W przypadku powiązań związanych z eksportami lub importami użytkownik może określić sposób obsługi błędów występujących podczas przetwarzania.

Użytkownik określa w powiązaniach również informacje specyficzne dla transportu. Na przykład dla powiązania HTTP określany jest adres URL punktu końcowego. Informacje specyficzne dla transportu, dotyczące powiązania HTTP, są opisane w tematach “Generowanie powiązania importu HTTP” i “Generowanie powiązania eksportu HTTP”. W Centrum informacyjnym można również znaleźć informacje o innych powiązaniach.

Transformacja formatu danych w importach i eksportach:

Kiedy w produkcie IBM Integration Designer konfigurowane jest powiązanie eksportu lub importu, jedną z określanych właściwości konfiguracji jest format danych używany przez powiązanie.

- W przypadku powiązań eksportu, w których aplikacja kliencka wysyła żądania do komponentu SCA i odbiera odpowiedzi z tego komponentu, użytkownik wskazuje format danych rodzimych. W zależności od tego formatu system wybiera odpowiednią procedurę obsługi danych lub odpowiednie powiązanie danych, aby transformować dane rodzime na obiekt biznesowy (używany przez komponent SCA), a także aby transformować obiekt biznesowy na dane rodzime (będące odpowiedzią dla aplikacji klienckiej).
- W przypadku powiązań importu, w których komponent SCA wysyła żądania do usługi poza modułem i odbiera odpowiedzi z tej usługi, użytkownik wskazuje format danych dla danych rodzimych. W zależności od tego formatu system wybiera odpowiednią procedurę obsługi danych lub odpowiednie powiązanie danych, aby transformować obiekt biznesowy na dane rodzime i odwrotnie.

IBM Business Process Manager udostępnia zbiór predefiniowanych formatów danych i odpowiadających im procedur obsługi danych lub powiązań danych, które obsługują te formaty. Użytkownik może również utworzyć własne niestandardowe procedury obsługi danych i zarejestrować format danych dla tych procedur obsługi danych. Więcej informacji zawiera temat “Tworzenie procedur obsługi danych” w Centrum informacyjnym produktu IBM Integration Designer.

- *Procedury obsługi danych* są niezależne od protokołu i transformują dane z jednego formatu na inny. W produkcie IBM Business Process Manager procedury obsługi danych zazwyczaj transformują dane rodzime (na przykład XML, CSV lub COBOL) na obiekt biznesowy oraz obiekt biznesowy na dane rodzime. Ponieważ procedury są niezależne od protokołu, można ponownie wykorzystywać tę samą procedurę obsługi danych z różnymi powiązaniem eksportu i importu. Można na przykład używać tej samej procedury obsługi danych XML z powiązaniem eksportu lub importu HTTP albo z powiązaniem eksportu lub importu JMS.
- *Powiązania danych* również transformują dane rodzime na obiekty biznesowe (i odwrotnie), ale są one specyficzne dla protokołu. Na przykład powiązania danych HTTP można użyć tylko z powiązaniem importu lub eksportu HTTP. W przeciwieństwie do procedur obsługi danych, powiązania danych HTTP nie można ponownie wykorzystać z powiązaniem importu lub eksportu MQ.

Uwaga: W produkcie IBM Business Process Manager 7.0 trzy powiązania danych HTTP (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML i HTTPServiceGatewayDataBinding) są nieaktualne. Zamiast nich należy używać procedur obsługi danych (jeśli to możliwe).

Jak wspomniano wcześniej, użytkownik może w razie potrzeby tworzyć niestandardowe procedury obsługi danych. Istnieje również możliwość utworzenia niestandardowych powiązań danych, ale zaleca się tworzenie niestandardowych procedur obsługi danych, ponieważ mogą one być używane z wieloma powiązaniem.

Procedury obsługi danych:

Procedury obsługi danych są konfigurowane w połączeniu z powiązaniem eksportu i importu w celu transformowania danych z jednego formatu na inny w sposób niezależny od protokołu. W ramach produktu udostępnianych jest kilka procedur obsługi danych, ale użytkownik może również w razie potrzeby utworzyć własną procedurę obsługi danych. Procedury obsługi danych można powiązać z powiązaniem eksportu lub powiązaniem importu na jednym z dwóch poziomów. Można je powiązać ze wszystkimi operacjami w interfejsie eksportu lub importu albo z konkretną operacją dla żądania lub odpowiedzi.

Predefiniowane procedury obsługi danych

Do określania procedury obsługi danych, która ma być wykorzystywana, służy produkt IBM Integration Designer.

Predefiniowane procedury obsługi danych są wymienione w poniższej tabeli, która opisuje również sposób transformowania danych przychodzących i wychodzących przez procedurę obsługi danych.

Uwaga: Te procedury obsługi danych mogą być używane z powiązaniem JMS, ogólnymi powiązaniem JMS, powiązaniem MQ JMS, powiązaniem WebSphere MQ i powiązaniem HTTP (z wyjątkiem wskazanych przypadków, w których nie można ich użyć).

Szczegółowe informacje zawiera temat “Procedury obsługi danych” w Centrum informacyjnym produktu Integration Designer.

Tabela 23. Predefiniowane procedury obsługi danych

Procedura obsługi danych	Dane rodzime na obiekt biznesowy	Obiekt biznesowy na dane rodzime
ATOM	Analizuje kanały informacyjne ATOM i umieszcza dane w obiekcie biznesowym kanału informacyjnego ATOM.	Serializuje obiekt biznesowy kanału informacyjnego ATOM do danych w kanale informacyjnym ATOM.
Z ogranicznikami	Analizuje dane z ogranicznikami i umieszcza je w obiekcie biznesowym.	Serializuje obiekt biznesowy do postaci danych z ogranicznikami (między innymi CSV).
O stałej długości	Analizuje dane o stałej długości i umieszcza je w obiekcie biznesowym.	Serializuje obiekt biznesowy do danych o stałej długości.
Obsługiwana przez produkt WTX	Deleguje transformację formatu danych do produktu WebSphere Transformation Extender (WTX). Nazwa odwzorowania WTX jest określana przez procedurę obsługi danych.	Deleguje transformację formatu danych do produktu WebSphere Transformation Extender (WTX). Nazwa odwzorowania WTX jest określana przez procedurę obsługi danych.
Obsługiwana przez użytkownika wywołującego produkt WTX	Deleguje transformację formatu danych do produktu WebSphere Transformation Extender (WTX). Nazwa odwzorowania WTX jest podawana przez użytkownika.	Deleguje transformację formatu danych do produktu WebSphere Transformation Extender (WTX). Nazwa odwzorowania WTX jest podawana przez użytkownika.
JAXB	Serializuje komponenty bean języka Java do postaci obiektu biznesowego przy użyciu reguł odwzorowywania zdefiniowanych przez specyfikację Java Architecture for XML Binding (JAXB).	Deserializuje obiekt biznesowy w komponenty bean języka Java przy użyciu reguł odwzorowywania zdefiniowanych przez specyfikację JAXB.
JAXWS Uwaga: Procedura obsługi danych JAXWS może być używana tylko z powiązaniem EJB.	Procedura używana przez powiązanie EJB do transformowania obiektu Java odpowiedzi lub obiektu wyjątku Java w obiekt biznesowy odpowiedzi przy użyciu reguł odwzorowywania zdefiniowanych przez specyfikację Java API for XML Web Services JAX-WS.	Procedura używana przez powiązanie EJB do transformowania obiektu biznesowego na parametry wychodzącej metody Java przy użyciu reguł odwzorowywania zdefiniowanych przez specyfikację JAX-WS.
JSON	Analizuje dane JSON i umieszcza je w obiekcie biznesowym.	Serializuje obiekt biznesowy do danych JSON.
Rodzima treść	Analizuje rodzime bajty, tekst, odwzorowanie, strumień lub obiekt i umieszcza dane w jednym z pięciu podstawowych obiektów biznesowych (tekstu, bajtów, odwzorowania, strumienia lub obiektu).	Transformuje pięć podstawowych obiektów biznesowych w bajty, tekst, odwzorowanie, strumień lub obiekt.
SOAP	Analizuje komunikat SOAP (oraz nagłówek) i umieszcza dane w obiekcie biznesowym.	Serializuje obiekt biznesowy do komunikatu SOAP.
XML	Analizuje dane XML i umieszcza je w obiekcie biznesowym.	Serializuje obiekt biznesowy do danych XML.
Procedura obsługi danych UTF8 XML	Analizuje dane XML kodowane przy użyciu kodowania UTF-8 i umieszcza je w obiekcie biznesowym.	Serializuje obiekt biznesowy do postaci danych XML kodowanych przy użyciu kodowania UTF-8 podczas wysyłania komunikatu.

Tworzenie procedury obsługi danych

Szczegółowe informacje o tworzeniu procedury obsługi danych zawiera temat “Tworzenie procedur obsługi danych” w Centrum informacyjnym produktu Integration Designer.

Powiązania danych:

Powiązania danych są konfigurowane w połączeniu z powiązaniem eksportu i importu w celu transformowania danych z jednego formatu na inny. Powiązania danych są specyficzne dla protokołu. W ramach produktu udostępnianych jest kilka powiązań danych, ale użytkownik może również w razie potrzeby utworzyć własne powiązanie danych. Powiązanie danych można powiązać z powiązaniem eksportu lub powiązaniem importu na jednym z dwóch poziomów. Powiązać je można ze wszystkimi operacjami interfejsu eksportu lub z konkretną operacją żądania lub odpowiedzi.

Do określenia powiązania danych, które ma być wykorzystywane albo do utworzenia własnego powiązania danych należy użyć produktu IBM Integration Designer. Opis procesu tworzenia powiązań danych zawiera sekcja “Przegląd powiązań JMS, MQ JMS i ogólnych powiązań JMS” w Centrum informacyjnym produktu IBM Integration Designer.

Powiązania JMS

Poniższa tabela przedstawia powiązania danych, których można użyć z następującymi powiązaniem:

- Powiązania JMS
- ogólne powiązania JMS,
- powiązania WebSphere MQ JMS.

Tabela zawiera również opis zadań wykonywanych przez powiązania danych.

Tabela 24. Predefiniowane powiązania danych dla powiązań JMS

Powiązanie danych	Dane rodzime na obiekt biznesowy	Obiekt biznesowy na dane rodzime
Serializowany obiekt Java	Transformuje serializowany obiekt Java w obiekt biznesowy (który jest odwzorowany jako typ wejściowy lub typ wyjściowy w pliku WSDL).	Przekształca obiekt biznesowy do serializowanego obiektu Java w komunikacie obiektu JMS.
Opakowane bajty	Wyodrębnia bajty z przychodzącego komunikatu bajtów JMS i opakowuje je w obiekt biznesowy JMSByteBuffer.	Wyodrębnia bajty z obiektu biznesowego JMSByteBuffer i opakowuje je w wychodzący komunikat bajtów JMS.
Opakowana pozycja odwzorowania	Wyodrębnia informacje o nazwie, wartości i typie dla każdej pozycji w przychodzącym komunikacie odwzorowania JMS i tworzy listę obiektów biznesowych MapEntry. Następnie opakowuje listę w obiekt biznesowy JMSMapBody.	Wyodrębnia informacje o nazwie, wartości i typie z listy MapEntry w obiekcie biznesowym JMSMapBody i tworzy odpowiednie pozycje w wychodzącym komunikacie odwzorowania JMS.
Opakowany obiekt	Wyodrębnia obiekt z przychodzącego komunikatu obiektu JMS i opakowuje go w obiekt biznesowy JMSEntityBody.	Wyodrębnia obiekt z obiektu biznesowego JMSEntityBody i opakowuje go w wychodzący komunikat obiektu JMS.
Opakowany tekst	Wyodrębnia tekst z przychodzącego komunikatu tekstu JMS i opakowuje go w obiekt biznesowy JMSTextBody.	Wyodrębnia tekst z obiektu biznesowego JMSTextBody i opakowuje go w wychodzący komunikat tekstu JMS.

Powiązania produktu WebSphere MQ

Poniższa tabela przedstawia powiązania danych, których można użyć z powiązaniem WebSphere MQ oraz opisuje czynności wykonywane przez te powiązania danych.

Tabela 25. Predefiniowane powiązania danych dla powiązań WebSphere MQ

Powiązanie danych	Dane rodzime na obiekt biznesowy	Obiekt biznesowy na dane rodzime
Serializowany obiekt Java	Transformuje serializowany obiekt Java pochodzący z komunikatu przychodzącego na obiekt biznesowy (który jest odwzorowany jako typ wejściowy lub typ wyjściowy w pliku WSDL).	Transformuje obiekt biznesowy w serializowany obiekt Java umieszczony w komunikacie wychodzącym.
Opakowane bajty	Wyodrębnia bajty z nieustrukturyzowanego komunikatu bajtowego MQ i opakowuje je w obiekt biznesowy JMSBytesBody.	Wyodrębnia bajty z obiektu biznesowego JMSBytesBody i opakowuje bajty w wychodzący nieustrukturyzowany komunikat bajtowy MQ.
Opakowany tekst	Wyodrębnia tekst z nieustrukturyzowanego komunikatu tekstu MQ i opakowuje go w obiekt biznesowy JMSTextBody.	Wyodrębnia tekst z obiektu biznesowego JMSTextBody i opakowuje go w nieustrukturyzowany komunikat tekstu MQ.
Opakowana pozycja strumienia	Wyodrębnia informacje o nazwie i typie dla każdej pozycji w przychodzącym komunikacie strumienia JMS i tworzy listę obiektów biznesowych StreamEntry. Następnie opakowuje listę w obiekt biznesowy JMSStreamBody.	Wyodrębnia informacje o nazwie i typie z listy StreamEntry w obiekcie biznesowym JMSStreamBody i tworzy odpowiednie pozycje w wychodzącym komunikacie JMSStreamMessage.

Oprócz powiązań danych wymienionych w tabeli Tabela 25 produkt WebSphere MQ używa również powiązań danych nagłówka. Szczegółowe informacje na ten temat zawiera Centrum informacyjne produktu IBM Integration Designer.

Powiązania HTTP

Poniższa tabela przedstawia powiązania danych, których można użyć z powiązaniem HTTP, oraz opisuje zadania wykonywane przez powiązania danych.

Tabela 26. Predefiniowane powiązania danych dla powiązań HTTP

Powiązanie danych	Dane rodzime na obiekt biznesowy	Obiekt biznesowy na dane rodzime
Opakowane bajty	Wyodrębnia bajty z treści przychodzącego komunikatu HTTP i opakowuje je w obiekt biznesowy HTTPBytes.	Wyodrębnia bajty z obiektu biznesowego HTTPBytes i dodaje je do treści wychodzącego komunikatu HTTP.
Opakowany tekst	Wyodrębnia tekst z treści przychodzącego komunikatu HTTP i opakowuje go w obiekt biznesowy HTTPText.	Wyodrębnia tekst z obiektu biznesowego HTTPText i dodaje go do treści wychodzącego komunikatu HTTP.

Selektory funkcji w powiązaniach eksportu:

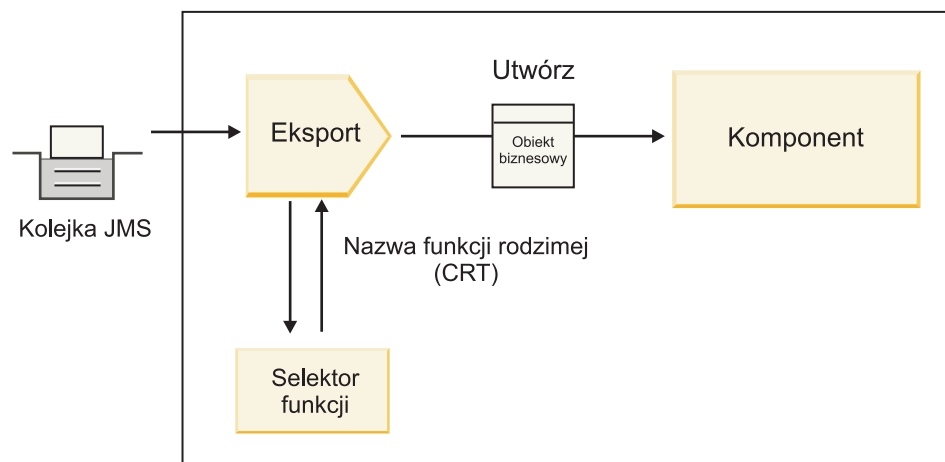
Selektor funkcji służy do wskazania, która operacja powinna zostać wykonana na danych dla określonego komunikatu żądania. Selektory funkcji są konfigurowane w ramach powiązania eksportu.

Rozważmy eksport SCA, który ujawnia interfejs. Interfejs zawiera dwie operacje: tworzenia i aktualizacji. Eksport ma powiązanie JMS, które odczytuje dane z kolejki.

Kiedy komunikat dochodzi do kolejki, do eksportu przekazywane są powiązane dane, ale jak zdecydować, która operacja z interfejsu eksportu powinna być wywołana dla połączonego komponentu? Operacja jest określana przy użyciu selektora funkcji oraz konfiguracji powiązania eksportu.

Selektor funkcji zwraca rodzimą nazwę funkcji (nazwę funkcji w systemie klienta wysyłającym komunikat). Rodzima nazwa funkcji jest następnie odwzorowywana na nazwę operacji lub funkcji w interfejsie powiązonym z eksportem. Na

poniższej przykładowej ilustracji selektor funkcji zwraca rodzimą nazwę funkcji (CRT) z komunikatu przychodzącego, następnie rodzima nazwa funkcji jest odwzorowywana na operację tworzenia, a obiekt biznesowy jest wysyłany do komponentu SCA z operacją tworzenia.



Rysunek 7. Selektor funkcji

Jeśli interfejs ma tylko jedną operację, nie ma potrzeby określania selektora funkcji.

Produkt udostępnia kilka wstępnie spakowanych selektorów funkcji, które są wymienione w kolejnych sekcjach.

Powiązania JMS

Poniższa tabela przedstawia selektory funkcji, których można użyć z następującymi powiązaniem:

- powiązania JMS,
- ogólne powiązania JMS,
- powiązania JMS produktu WebSphere MQ.

Tabela 27. Predefiniowane selektory funkcji dla powiązań JMS

Selektor funkcji	Opis
Selektor funkcji JMS dla prostych powiązań danych JMS	Wybiera nazwę operacji na podstawie właściwości JMSType komunikatu.
Selektor funkcji właściwości nagłówka JMS	Zwraca z nagłówka wartość właściwości łańcucha JMS TargetFunctionName.
Selektor funkcji bramy usług JMS	Określa, czy żądanie jest operacją jednokierunkową czy dwukierunkową, badając właściwość JMSReplyTo ustawioną przez klienta.

Powiązania produktu WebSphere MQ

Poniższa tabela przedstawia selektory funkcji, których można użyć z powiązaniem WebSphere MQ.

Tabela 28. Predefiniowane selektory funkcji dla powiązań WebSphere MQ

Selektor funkcji	Opis
Selektor funkcji handleMessage produktu MQ	Zwraca funkcję handleMessage w postaci wartości, która jest odwzorowywana przy użyciu powiązań metody eksportu na nazwę operacji w interfejsie.

Tabela 28. Predefiniowane selektory funkcji dla powiązań WebSphere MQ (kontynuacja)

Selektor funkcji	Opis
Produkt MQ używa domyślnego selektora funkcji JMS	Odczytuje rodzimą operację z właściwości TargetFunctionName folderu nagłówka MQRFH2.
Produkt MQ używa formatu treści komunikatu jako funkcji rodzimej	Znajduje pole Format ostatniego nagłówka i zwraca to pole w postaci łańcucha.
Selektor funkcji type produktu MQ	Tworzy metodę w powiązaniu eksportu poprzez pobranie adresu URL zawierającego właściwości Msd, Set, Type i Format znajdujące się w nagłówku MQRFH2.
Selektor funkcji bramy usług MQ	Używa właściwości MsgType w nagłówku MQMD do określenia nazwy operacji.

Powiązania HTTP

Poniższa tabela przedstawia selektory funkcji, których można użyć z powiązaniem HTTP.

Tabela 29. Predefiniowane selektory funkcji dla powiązań HTTP

Selektor funkcji	Opis
Selektor funkcji HTTP oparty na nagłówku TargetFunctionName	Używa właściwości nagłówka HTTP TargetFunctionName klienta, aby określić operację, która ma być wywoływana z eksportu w czasie wykonywania.
Selektor funkcji HTTP oparty na metodzie adresu URL i protokołu HTTP	Używa ścieżki względnej z adresu URL, do której dopisana jest metoda HTTP z klienta, aby określić rodzimą operację zdefiniowaną w eksporcie.
Selektor funkcji bramy usług HTTP oparty na metodzie adresu URL z nazwą operacji	Określa metodę wywołania na podstawie adresu URL, jeśli do adresu URL żądania został dopisany łańcuch „operationMode = oneway”.

Uwaga: Istnieje również możliwość utworzenia własnego selektora funkcji przy użyciu produktu IBM Integration Designer. Informacje na temat tworzenia selektora funkcji zawiera Centrum informacyjne produktu IBM Integration Designer. Na przykład sekcja “Przegląd selektorów funkcji MQ” zawiera opis tworzenia selektora funkcji dla powiązań WebSphere MQ.

Obsługa błędów:

Użytkownik może skonfigurować powiązania importu i eksportu tak, aby obsługiwać błędy (na przykład wyjątki biznesowe) występujące podczas przetwarzania. Jest to realizowane poprzez określenie procedur obsługi danych błędu. Procedura obsługi danych błędu może być skonfigurowana na trzech poziomach. Procedurę obsługi danych błędu można powiązać z błędem, z operacją lub ze wszystkimi operacjami za pomocą powiązania.

Procedura obsługi danych błędu przetwarza dane błędu i transformuje je na poprawny format w celu wysłania przez powiązanie eksportu lub importu.

- W przypadku powiązania eksportu procedura obsługi danych błędu transformuje obiekt biznesowy wyjątku wysłany z komponentu na komunikat odpowiedzi, który może być używany przez aplikację kliencką.
- W przypadku powiązania importu procedura obsługi danych błędu transformuje dane błędu lub komunikat odpowiedzi wysłany z usługi na obiekt biznesowy wyjątku, który może być użyty przez komponent SCA.

W przypadku powiązań importu powiązanie wywołuje selektor błędu, który określa czy komunikat odpowiedzi jest normalną odpowiedzią, niepowodzeniem biznesowym czy wyjątkiem czasu wykonywania.

Istnieje możliwość określenia procedury obsługi danych błędu dla konkretnego błędu, dla operacji lub dla wszystkich operacji powiązania.

- Jeśli procedura obsługi danych błędu jest skonfigurowana na wszystkich trzech poziomach, wywoływana jest procedura obsługi danych powiązana z określonym błędem.
- Jeśli procedury obsługi danych błędu są skonfigurowane na poziomie operacji i powiązania, wywoływana jest procedura obsługi danych powiązana z operacją.

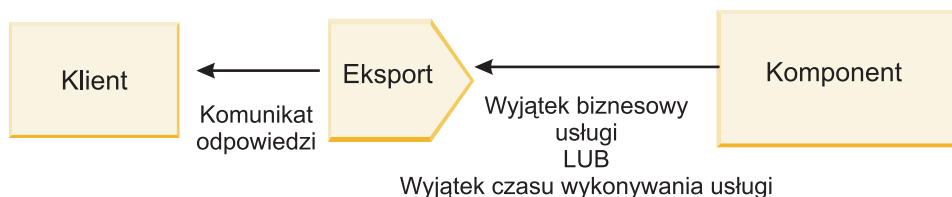
Do określania sposobu obsługi błędów używane są dwa edytory w produkcie IBM Integration Designer. Edytor interfejsów służy do wskazania, czy w operacji będzie występował błąd. Po wygenerowaniu powiązania z tym interfejsem edytor w widoku właściwości umożliwia skonfigurowanie sposobu obsługi błędu. Więcej informacji zawiera temat “Selektory błędów” w Centrum informacyjnym produktu IBM Integration Designer.

Sposób obsługi błędów w powiązaniach eksportu:

Jeśli podczas przetwarzania żądania z aplikacji klienckiej wystąpi błąd, powiązanie eksportu może zwrócić informacje o błędzie klientowi. Użytkownik konfiguruje powiązanie eksportu, określając sposób przetwarzania błędu oraz sposób jego zwracania klientowi.

Powiązanie eksportu jest konfigurowane przy użyciu produktu IBM Integration Designer.

Podczas przetwarzania żądania klient wywołuje eksport przy użyciu żądania, a eksport wywołuje komponent SCA. Podczas przetwarzania żądania komponent SCA może zwrócić odpowiedź biznesową lub zgłosić wyjątek biznesowy usługi albo wyjątek czasu wykonywania usługi. W takiej sytuacji powiązanie eksportu transformuje wyjątek na komunikat błędu i wysyła ten komunikat do klienta, jak przedstawiono na poniższej ilustracji i jak opisano w kolejnych sekcjach.



Rysunek 8. Sposób przesyłania informacji o błędzie z komponentu, przez powiązanie eksportu, do klienta

Użytkownik może utworzyć niestandardową procedurę obsługi danych lub niestandardowe powiązanie danych do obsługi błędów.

Niepowodzenia biznesowe

Niepowodzenia biznesowe są błędami lub wyjątkami biznesowymi, które występują podczas przetwarzania.

Rozważmy następujący interfejs, na którym wykonywana jest operacja createCustomer (tworzenie klienta). Ta operacja ma zdefiniowane dwa niepowodzenia biznesowe: CustomerAlreadyExists (klient już istnieje) i MissingCustomerId (brak identyfikatora klienta).

▼ Operacje

Operacje i ich parametry

	Nazwa	Typ
▼ createCustomer		
▶ Dane wejściowe	Dane wejściowe	Informacje o kliencie
▶ Dane wyjściowe	Dane wyjściowe	Informacje o kliencie
✖ Błąd	Klient już istnieje	Obiekt biznesowy - klient już istnieje
✖ Błąd	Brak identyfikatora klienta	Obiekt biznesowy - brak identyfikatora klienta

Rysunek 9. Interfejs z dwoma błędami

W tym przykładzie, jeśli klient wysłał żądanie utworzenia klienta (do tego komponentu SCA), a ten klient już istnieje, komponent zgłasza błąd `CustomerAlreadyExists` do eksportu. Eksport musi propagować to niepowodzenie biznesowe z powrotem do klienta wywołującego. W tym celu używa procedury obsługi danych błędów, która jest skonfigurowana w powiązaniu eksportu.

Po odebraniu niepowodzenia biznesowego przez powiązanie eksportu, realizowane są następujące działania przetwarzania:

1. Powiązanie określa, która procedura obsługi danych błędów ma być wywołana w celu obsługi błędów. Jeśli wyjątek biznesowy usługi zawiera nazwę błędów, wywoływana jest procedura obsługi danych błędów skonfigurowana dla błędów. Jeśli wyjątek biznesowy usługi nie zawiera nazwy błędów, nazwa błędów jest uzyskiwana poprzez dopasowywanie typów błędów.
2. Powiązanie wywołuje procedurę obsługi danych błędów przy użyciu obiektu danych z wyjątku biznesowego usługi.
3. Procedura obsługi danych błędów transformuje obiekt danych błędów na komunikat odpowiedzi i zwraca ten komunikat do powiązania eksportu.
4. Eksport zwraca komunikat odpowiedzi klientowi.

Jeśli wyjątek biznesowy usługi zawiera nazwę błędów, wywoływana jest procedura obsługi danych błędów skonfigurowana dla błędów. Jeśli wyjątek biznesowy usługi nie zawiera nazwy błędów, nazwa błędów jest uzyskiwana poprzez dopasowywanie typów błędów.

Wyjątki czasu wykonywania

Wyjątek czasu wykonywania jest wyjątkiem, który występuje w aplikacji SCA podczas przetwarzania żądania nieodpowiadającego niepowodzeniu biznesowemu. W przeciwieństwie do niepowodzeń biznesowych, wyjątki czasu wykonywania nie są zdefiniowane w interfejsie.

W niektórych scenariuszach użytkownik może propagować te wyjątki czasu wykonywania do aplikacji klienckiej, aby aplikacja ta mogła podjąć odpowiednie działania.

Jeśli na przykład klient wysłał żądanie (do komponentu SCA), aby utworzyć klienta, a podczas przetwarzania żądania wystąpi błąd autoryzacji, komponent zgłasza wyjątek czasu wykonywania. Ten wyjątek czasu wykonywania musi być propagowany z powrotem do klienta wywołującego, aby klient ten podjął odpowiednie działanie dotyczące autoryzacji. Jest to realizowane przez procedurę obsługi danych wyjątku czasu wykonywania skonfigurowaną w powiązaniu eksportu.

Uwaga: Procedurę obsługi danych wyjątku czasu wykonywania można skonfigurować tylko dla powiązań HTTP.

Przetwarzanie wyjątku czasu wykonywania jest podobne do przetwarzania niepowodzenia biznesowego. Jeśli skonfigurowana została procedura obsługi danych wyjątku czasu wykonywania, realizowane są następujące działania przetwarzania:

1. Powiązanie eksportu wywołuje odpowiednią procedurę obsługi danych przy użyciu wyjątku czasu wykonywania usługi.
2. Procedura obsługi danych transformuje obiekt danych błędu na komunikat odpowiedzi i zwraca ten komunikat do powiązania eksportu.
3. Eksport zwraca komunikat odpowiedzi klientowi.

Obsługa błędów i obsługa wyjątków czasu wykonywania są działaniami opcjonalnymi. Jeśli użytkownik nie chce propagować błędów lub wyjątków czasu wykonywania do klienta wywołującego, nie należy konfigurować procedury obsługi danych błędu lub procedury obsługi danych wyjątku czasu wykonywania.

Sposób obsługi błędów w powiązaniach importu:

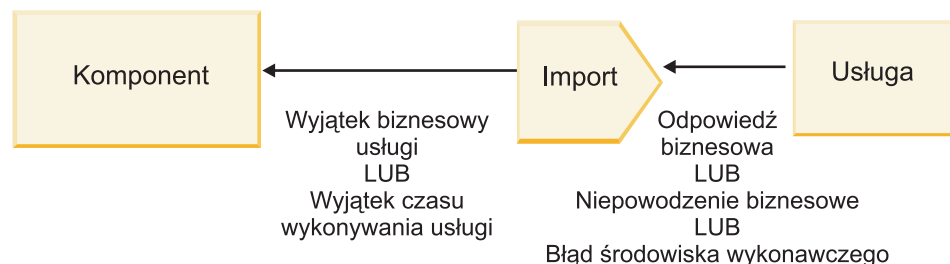
Komponent używa importu, aby wysłać żądanie do usługi znajdującej się poza modulem. Jeśli podczas przetwarzania żądania wystąpi błąd, usługa zwraca błąd do powiązania importu. Użytkownik może skonfigurować powiązanie importu, aby określić sposób przetwarzania błędu oraz sposób jego zwracania komponentowi.

Powiązania importu są konfigurowane przy użyciu produktu IBM Integration Designer. Istnieje możliwość określenia procedury obsługi danych błędu (lub powiązania danych błędu) i selektora błędu.

Procedury obsługi danych błędu

Usługa przetwarzająca żądanie wysyła do powiązania importu informacje o błędzie w postaci wyjątku lub komunikatu odpowiedzi zawierającego dane błędu.

Powiązanie importu transformuje wyjątek usługi lub komunikat odpowiedzi na wyjątek biznesowy usługi lub wyjątek czasu wykonywania usługi, jak przedstawiono na poniższej ilustracji i jak opisano w kolejnych sekcjach.



Rysunek 10. Sposób przesyłania informacji o błędzie z usługi, przez import, do komponentu

Użytkownik może utworzyć niestandardową procedurę obsługi danych lub niestandardowe powiązanie danych do obsługi błędów.

Selektory błędów

Podczas konfigurowania powiązania importu można określić selektor błędu. Selektor błędu określa, czy odpowiedź importu jest rzeczywistą odpowiedzią, wyjątkiem biznesowym czy błędem czasu wykonywania. Określa również, na podstawie treści odpowiedzi lub nagłówka, rodzimą nazwę błędu, która jest odwzorowywana przez konfigurację powiązania na nazwę błędu w powiązonym interfejsie.

Z importami JMS, importami MQ JMS, ogólnymi importami JMS, importami WebSphere MQ i importami HTTP mogą być używane dwa typy wstępnie spakowanych selektorów błędów.

Tabela 30. Wstępnie spakowane selektory błędów

Typ selektora błędu	Opis
Oparty na nagłówku	Określa, czy komunikat odpowiedzi jest niepowodzeniem biznesowym, wyjątkiem czasu wykonywania, czy normalnym komunikatem, na podstawie nagłówków w przychodzącym komunikacie odpowiedzi.
SOAP	Określa czy komunikat odpowiedzi SOAP jest normalną odpowiedzią, niepowodzeniem biznesowym, czy wyjątkiem czasu wykonywania.

Poniżej przedstawiono przykłady selektorów błędów opartych na nagłówkach oraz selektora błędu SOAP.

- Selektor błędu oparty na nagłówkach

Jeśli aplikacja ma wskazać, że komunikat przychodzący jest niepowodzeniem biznesowym, w komunikacie przychodzącym muszą występować dwa następujące nagłówki dla niepowodzeń biznesowych:

Header name = FaultType, Header value = Business

Header name = FaultName, Header value = <zdefiniowana przez użytkownika rodzima nazwa błędu>

Jeśli aplikacja ma wskazać, że przychodzący komunikat odpowiedzi jest wyjątkiem czasu wykonywania, w komunikacie przychodzącym musi występować jeden następujący nagłówek:

Header name = FaultType, Header value = Runtime

- Selektor błędów SOAP

Niepowodzenie biznesowe może być wysłane w ramach komunikatu SOAP przy użyciu następującego niestandardowego nagłówka SOAP. W tym przypadku nazwą błędu jest CustomerAlreadyExists.

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

Selektor błędu jest opcjonalny. Jeśli użytkownik nie określi selektora błędu, powiązanie importu nie może określić typu odpowiedzi. W takim przypadku powiązanie traktuje ją jako odpowiedź biznesową i wywołuje procedurę obsługi danych odpowiedzi lub powiązanie danych odpowiedzi.

Istnieje możliwość utworzenia niestandardowego selektora błędu. Kroki tworzenia niestandardowego selektora błędu są przedstawione w temacie “Tworzenie niestandardowego selektora błędu” w Centrum informacyjnym produktu IBM Integration Designer.

Niepowodzenia biznesowe

Niepowodzenie biznesowe może wystąpić w sytuacji, kiedy występuje błąd przetwarzania żądania. Jeśli na przykład użytkownik wysłał żądanie utworzenia klienta, a ten klient już istnieje, usługa wysłała wyjątek biznesowy do powiązania importu.

Kroki przetwarzania wykonywane po odebraniu wyjątku biznesowego przez powiązanie są zależne od tego, czy dla powiązania został skonfigurowany selektor błędu.

- Jeśli nie został skonfigurowany selektor błędu, powiązanie wywołuje procedurę obsługi danych odpowiedzi lub powiązanie danych odpowiedzi.
- Jeśli został skonfigurowany selektor błędu, realizowane są następujące działania przetwarzania:
 1. Powiązanie importu wywołuje selektor błędu, aby określić, czy odpowiedź jest niepowodzeniem biznesowym, odpowiedzią biznesową czy błędem czasu wykonywania.
 2. Jeśli odpowiedź jest niepowodzeniem biznesowym, powiązanie importu wywołuje selektor błędu, aby udostępnić rodzimą nazwę błędu.
 3. Powiązanie importu określa błąd WSDL odpowiadający rodzimej nazwie błędu zwróconej przez selektor błędu.
 4. Powiązanie importu określa procedurę obsługi danych błędu skonfigurowaną dla tego błędu WSDL.

5. Powiązanie importu wywołuje tę procedurę obsługi danych błędu przy użyciu danych błędu.
6. Procedura obsługi danych błędu transformuje dane błędu na obiekt danych i zwraca ten obiekt do powiązania importu.
7. Powiązanie importu konstruuje obiekt wyjątku biznesowego usługi przy użyciu obiektu danych i nazwy błędu.
8. Import zwraca obiekt wyjątku biznesowego usługi do komponentu.

Wyjątki czasu wykonywania

Wyjątek czasu wykonywania może wystąpić, jeśli istnieje problem dotyczący komunikacji z usługą. Przetwarzanie wyjątku czasu wykonywania jest podobne do przetwarzania wyjątku biznesowego. Jeśli został skonfigurowany selektor błędu, realizowane są następujące działania przetwarzania:

1. Powiązanie importu wywołuje odpowiednią procedurę obsługi danych wyjątku czasu wykonywania przy użyciu danych wyjątku.
2. Procedura obsługi danych wyjątku czasu wykonywania transformuje dane wyjątku na obiekt wyjątku czasu wykonywania usługi i zwraca ten obiekt do powiązania importu.
3. Import zwraca obiekt wyjątku czasu wykonywania usługi do komponentu.

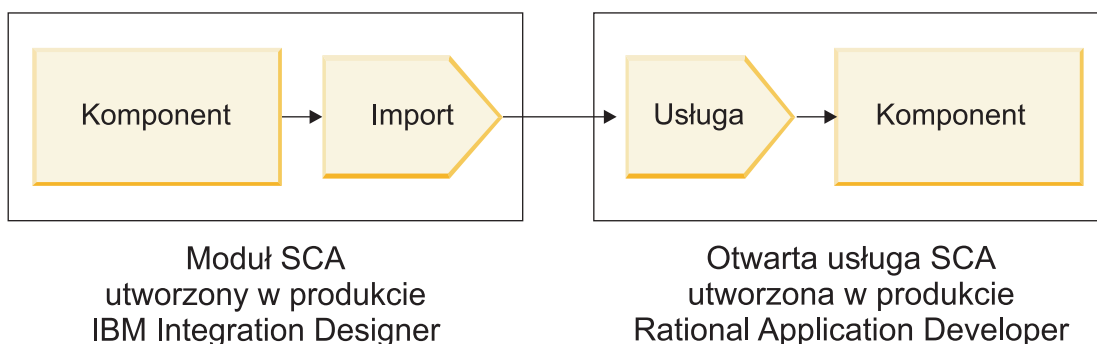
Współdziałanie między modułami SCA i usługami Open SCA

Pakiet składników IBM WebSphere Application Server 7.0 Feature Pack for Service Component Architecture (SCA) udostępnia prosty, ale jednocześnie bogaty w możliwości model programowania do konstruowania aplikacji opartych na specyfikacjach Open SCA. Moduły SCA produktu IBM Business Process Manager używają powiązań importu i eksportu do współdziałania z usługami Open SCA utworzonymi w środowisku Rational Application Developer i udostępnianymi przez pakiet składników WebSphere Application Server Feature Pack for Service Component Architecture.

Aplikacja SCA wywołuje aplikację Open SCA za pośrednictwem powiązania importu. Aplikacja SCA odbiera wywołanie z aplikacji Open SCA za pośrednictwem powiązania eksportu. Listę obsługiwanych powiązań zawiera sekcja “Wywoływanie usług za pośrednictwem powiązań współdziałających” na stronie 69.

Wywoływanie usług Open SCA z modułów SCA

Aplikacje SCA utworzone przy użyciu produktu IBM Integration Designer mogą wywoływać aplikacje Open SCA opracowane w środowisku Rational Application Developer. Niniejsza sekcja zawiera przykład wywoływania usługi Open SCA z poziomu modułu SCA przy użyciu powiązania importu SCA.



Rysunek 11. Komponent w module SCA wywołujący usługę Open SCA

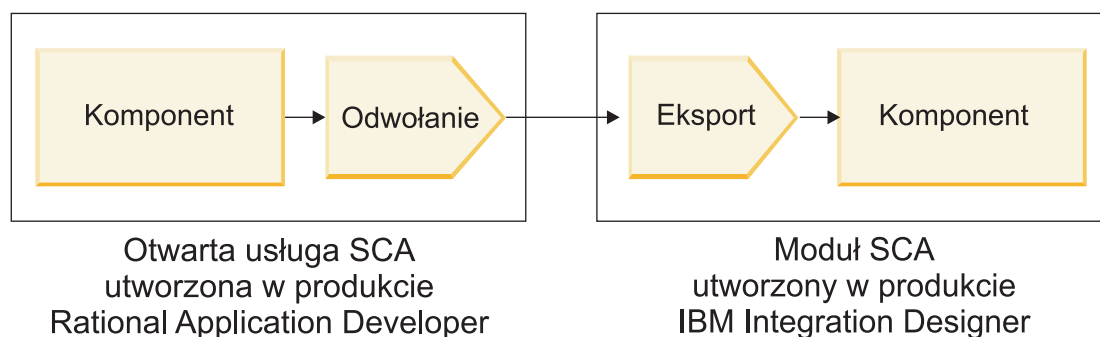
Do wywoływania usługi Open SCA nie jest wymagana żadna specjalna konfiguracja.

Aby połączyć się z usługą Open SCA za pośrednictwem powiązania importu SCA, użytkownik udostępnia w powiązaniu importu nazwę komponentu i nazwę usługi Open SCA.

1. Aby uzyskać nazwę komponentu docelowego i usługi z kompozytu Open SCA, wykonaj następujące kroki:
 - a. Upewnij się, czy otwarta jest karta **Właściwości**, wybierając opcję **Okna > Pokaż widok > Właściwości**.
 - b. Otwórz edytor kompozytów, klikając dwukrotnie diagram kompozytu zawierający komponent i usługę. Na przykład: diagramem kompozytu dla komponentu **klient** jest **klient.diagram_kompozytu**.
 - c. Kliknij komponent docelowy.
 - d. Zapisz nazwę komponentu docelowego wyświetlaną w polu **Nazwa** karty **Właściwości**.
 - e. Kliknij ikonę usługi powiązaną z komponentem.
 - f. Zapisz nazwę usługi wyświetlaną w polu **Nazwa** karty **Właściwości**.
2. Aby skonfigurować import produktu IBM Business Process Manager w celu połączenia go z usługą Open SCA, wykonaj następujące kroki:
 - a. W produkcie IBM Integration Designer przejdź do karty **Właściwości** dla importu SCA, który ma zostać połączony z usługą Open SCA.
 - b. W polu **Nazwa modułu** wprowadź nazwę komponentu zapisaną w kroku 1d.
 - c. W polu **Nazwa eksportu** wprowadź nazwę usługi zapisaną w kroku 1f.
 - d. Zapisz pracę, naciskając kombinację klawiszy Ctrl+S.

Wywoływanie modułów SCA z poziomu usług Open SCA

Aplikacje Open SCA utworzone w środowisku Rational Application Developer mogą wywoływać aplikacje SCA utworzone w produkcie IBM Integration Designer. Niniejsza sekcja zawiera przykład wywoływania modułu SCA (za pośrednictwem powiązania eksportu SCA) z poziomu usługi Open SCA.



Rysunek 12. Usługa Open SCA wywołująca komponent w module SCA

Aby nawiązać połączenie z komponentem SCA za pośrednictwem powiązania odwołania Open SCA, należy podać nazwę modułu i nazwę eksportu.

1. Aby uzyskać nazwę modułu docelowego i eksportu, wykonaj następujące kroki:
 - a. Otwórz moduł w edytorze składania produktu IBM Integration Designer, dwukrotnie klikając ten moduł.
 - b. Kliknij eksport.
 - c. Zapisz nazwę eksportu wyświetlaną w polu **Nazwa** karty **Właściwości**.
2. Skonfiguruj odwołanie Open SCA, które ma zostać połączone z modułem i eksportem produktu IBM Business Process Manager:
 - a. Otwórz edytor kompozytów w produkcie Rational Application Developer, dwukrotnie klikając diagram kompozytu zawierający komponent i usługę.
 - b. Kliknij ikonę odwołania dla odwołania komponentu, aby wyświetlić właściwości odwołania na karcie **Właściwości**.
 - c. Kliknij kartę **Powiązanie** umieszczoną po lewej stronie.
 - d. Kliknij opcję **Powiązania**, a następnie opcję **Dodaj**.

- e. Wybierz powiązanie **SCA**.
- f. W polu **Identyfikator URI** wprowadź nazwę modułu produktu IBM Business Process Manager, następnie ukośnik (“/”), a następnie nazwę eksportu (określoną w kroku 1c na stronie 68).
- g. Kliknij przycisk **OK**.
- h. Zapisz pracę, naciskając kombinację klawiszy Ctrl+S.

Wywoływanie usług za pośrednictwem powiązań współdziałających

W celu realizacji współdziałania z usługą Open SCA obsługiwane są następujące powiązania.

- Powiązanie SCA

Jeśli moduł SCA w produkcie IBM Business Process Manager wywołuje usługę Open SCA za pośrednictwem powiązania importu SCA, obsługiwane są następujące style wywoływania:

- asynchroniczny (jednokierunkowy),
- synchroniczny (żądanie-odpowiedź).

Interfejs importu SCA oraz interfejs usługi Open SCA muszą używać interfejsu WSDL zgodnego ze standardem współdziałania usług Web Services (Web Services Interoperability - WS-I).

Uwaga: Powiązanie SCA obsługuje propagację kontekstu transakcji i zabezpieczeń.

- Powiązanie usługi Web Service JAX-WS z protokołem SOAP1.1/HTTP lub SOAP1.2/HTTP

Interfejs importu SCA oraz interfejs usługi Open SCA muszą używać interfejsu WSDL zgodnego ze standardem współdziałania usług Web Services (Web Services Interoperability - WS-I).

Dodatkowo obsługiwane są następujące mechanizmy jakości usług:

- niepodzielne transakcje usług Web Services,
- zabezpieczenia usług Web Services.

- Powiązanie EJB

Do zdefiniowania interakcji między modułem SCA i usługą Open SCA w przypadku używania powiązania EJB stosowany jest interfejs Java.

Uwaga: powiązanie EJB obsługuje propagację kontekstu transakcji i zabezpieczeń.

- Powiązania JMS

Interfejs importu SCA oraz interfejs usługi Open SCA muszą używać interfejsu WSDL zgodnego ze standardem współdziałania usług Web Services (Web Services Interoperability - WS-I).

Obsługiwani są następujący dostawcy JMS:

- przesyłanie komunikatów na platformie WebSphere (powiązanie JMS),
- WebSphere MQ (powiązanie MQ JMS).

Uwaga: Grafy biznesowe nie są obejmowane współdziałaniem w ramach jakichkolwiek powiązań SCA, a więc nie są obsługiwane w interfejsach używanych do współdziałania z pakietem składników WebSphere Application Server Feature Pack for Service Component Architecture.

Typy powiązań

Powiązania specyficzne dla protokołu są używane z importami i eksportami w celu określenia sposobu transportowania danych do modułu lub z modułu.

Wybór odpowiednich powiązań:

Tworząc aplikację, należy wiedzieć, w jaki sposób wybrać powiązanie, które najlepiej odpowiada potrzebom danej aplikacji.

Powiązania dostępne w produkcie IBM Integration Designer oferują użytkownikowi kilka opcji do wyboru. Niniejsza lista pomaga określić typ powiązania najlepiej nadający się do realizacji potrzeb danej aplikacji.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Architektura Service Component Architecture (SCA)*:

- Wszystkie usługi są zawarte w modułach (czyli nie ma usług zewnętrznych).
- Użytkownik chce rozdzielić funkcje na różne moduły SCA, które bezpośrednio współdziałają ze sobą.
- Moduły są ściśle sprzężone.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć wykorzystanie powiązania *usługi Web Service*:

- Niezbędny jest dostęp do usługi zewnętrznej za pośrednictwem Internetu lub udostępnienie usługi przez Internet.
- Usługi są luźno sprzężone.
- Preferowana jest komunikacja synchroniczna, czyli żądanie z jednej usługi może oczekiwać na odpowiedź z innej.
- Protokołem usługi zewnętrznej do której uzyskiwany jest dostęp lub usługi, która ma zostać udostępniona, jest protokół SOAP/HTTP lub SOAP/JMS.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć wykorzystanie powiązania *HTTP*:

- Niezbędny jest dostęp do usługi zewnętrznej za pośrednictwem Internetu lub udostępnienie usługi przez Internet, a użytkownik pracuje z innymi usługami Web Services, takimi jak GET, PUT i DELETE.
- Usługi są luźno sprzężone.
- Preferowana jest komunikacja synchroniczna, czyli żądanie z jednej usługi może oczekiwać na odpowiedź z innej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Enterprise JavaBeans (EJB)*:

- Powiązanie jest przeznaczone dla zaimportowanej usługi, która sama jest komponentem EJB lub musi być dostępna za pośrednictwem klientów EJB.
- Zaimportowana usługa jest luźno sprzężona.
- Nie są wymagane stanowe interakcje EJB.
- Preferowana jest komunikacja synchroniczna, czyli żądanie z jednej usługi może oczekiwać na odpowiedź z innej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Enterprise Information Systems (EIS)*:

- Niezbędne jest uzyskanie dostępu do usługi w systemie EIS przy użyciu adaptera zasobu.
- Synchroniczna transmisja danych jest preferowana w odniesieniu do asynchronicznej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Java Message Service (JMS)*:

Ważne: Istnieje kilka typów powiązań JMS. Jeśli użytkownik oczekuje wymiany komunikatów SOAP za pośrednictwem JMS, należy rozważyć użycie powiązania usługi Web Service z protokołem SOAP/JMS. Patrz temat "Powiązania usług Web Services" na stronie 71.

- Niezbędne jest uzyskanie dostępu do systemu przesyłania komunikatów.
- Usługi są luźno sprzężone.
- Asynchroniczna transmisja danych jest preferowana w odniesieniu do synchronicznej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Generic Java Message Service (JMS)*:

- Niezbędne jest uzyskanie dostępu do systemu przesyłania komunikatów od dostawcy innego niż IBM.
- Usługi są luźno sprzężone.
- Niezawodność jest ważniejsza od wydajności, czyli asynchroniczna transmisja danych jest preferowana w odniesieniu do synchronicznej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Message Queue (MQ)*:

- Niezbędny jest dostęp do systemu przesyłania komunikatów WebSphere MQ oraz używania rodzimych funkcji produktu MQ.
- Usługi są luźno sprzężone.

- Niezawodność jest ważniejsza od wydajności, czyli asynchroniczna transmisja danych jest preferowana w odniesieniu do synchronicznej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć wykorzystanie powiązania *MQ JMS*:

- Niezbędne jest uzyskanie dostępu do systemu przesyłania komunikatów WebSphere MQ, ale może to być realizowane w kontekście JMS (czyli: podzbiór funkcji JMS jest wystarczający do realizacji potrzeb aplikacji).
- Usługi są luźno sprzężone.
- Niezawodność jest ważniejsza od wydajności, czyli asynchroniczna transmisja danych jest preferowana w odniesieniu do synchronicznej.

Powiązania SCA:

Powiązanie SCA (Service Component Architecture) umożliwia usługę komunikację z innymi usługami w innych modułach. Import z powiązaniem SCA pozwala uzyskać dostęp do usługi w innym module SCA. Eksport z powiązaniem SCA pozwala oferować usługę innym modułom.

Do generowania i konfigurowania powiązań SCA w importach i eksportach modułów SCA służy produkt IBM Integration Designer.

Jeśli moduły działają na tym samym serwerze lub są wdrożone w tym samym klastrze, powiązanie SCA jest najłatwiejszym i najszybszym powiązaniem, którego można użyć.

Po wdrożeniu modułu zawierającego powiązanie SCA na serwerze użytkownik może użyć Konsoli administracyjnej, aby wyświetlić informacje o powiązaniu lub, w przypadku powiązania importu, zmienić wybrane właściwości powiązania.

Powiązania usług Web Services:

Powiązanie usługi Web Service jest sposobem przesyłania komunikatów z komponentu SCA (Service Component Architecture) do usługi Web Service (i odwrotnie).

Przegląd powiązań usług Web Services: **Advanced**

Powiązanie importu usługi Web Service umożliwia wywoływanie zewnętrznej usługi Web Service z poziomu komponentów SCA (Service Component Architecture). Powiązanie eksportu usługi Web Service umożliwia ujawnianie klientom komponentów SCA w postaci usług Web Services.

Powiązanie usługi WWW umożliwia uzyskiwanie dostępu do usług zewnętrznych przy użyciu współdziałających komunikatów SOAP i funkcji jakości usług (QoS).

Do generowania i konfigurowania powiązań usług Web Services w importach i eksportach modułów SCA służy produkt Integration Designer. Dostępne są następujące typy powiązań usług Web Service:

- SOAP1.2/HTTP i SOAP1.1/HTTP

Te powiązania są oparte na interfejsie JAX-WS (Java API for XML Web Services) - interfejsie programowania języka Java służącym do tworzenia usług Web Services.

- Jeśli usługa Web Service jest zgodna ze specyfikacją SOAP 1.2, należy użyć powiązania SOAP1.2/HTTP.
- Jeśli usługa Web Service jest zgodna ze specyfikacją SOAP 1.1, należy użyć powiązania SOAP1.1/HTTP.

Ważne: Jeśli wdrażana jest aplikacja z powiązaniem usługi Web Service JAX-WS, docelowy serwer nie może mieć wybranej opcji **Uruchamiaj komponenty, gdy są potrzebne**. Szczegółowe informacje na ten temat zawiera sekcja "Sprawdzanie konfiguracji serwera" na stronie 80.

Po wybraniu jednego z tych powiązań użytkownik może wysłać załączniki z komunikatami SOAP.

Powiązania usług Web Services współpracują ze standardowymi komunikatami SOAP. Korzystając z jednego z powiązań usług Web Services JAX-WS, można jednak dostosować sposób analizowania lub zapisywania

komunikatów SOAP. Istnieje na przykład możliwość obsługi niestandardowych elementów w komunikatach SOAP lub zastosowania dodatkowego przetwarzania komunikatu SOAP. Podczas konfigurowania powiązania użytkownik określa niestandardową procedurę obsługi danych, która realizuje to przetwarzanie komunikatu SOAP.

Istnieje możliwość używania zestawów strategii z powiązaniem usługi Web Service JAX-WS. Zestaw strategii jest kolekcją typów strategii, z których każdy udostępnia jakość usług (QoS). Na przykład: zestaw strategii WSAddressing udostępnia niezależny od transportu sposób jednolitego adresowania usług Web Services i komunikatów. Do wybierania zestawu strategii na potrzeby powiązania służy produkt Integration Designer.

Uwaga: Aby używać zestawu strategii SAML (Security Assertion Markup Language), należy wykonać dodatkowe działania konfiguracyjne opisane w sekcji “Importowanie zestawów strategii SAML” na stronie 77.

- SOAP1.1/HTTP

Tego powiązania należy użyć, aby utworzyć usługi Web Services używające komunikatów kodowanych SOAP opartych na specyfikacji JAX-RPC (Java API for XML-based RPC).

- SOAP1.1/JMS

Tego powiązania należy użyć w celu wysyłania lub odbierania komunikatów SOAP przy użyciu miejsca docelowego JMS (Java Message Service).

Niezależnie od transportu używanego do przekazania komunikatów SOAP (HTTP lub JMS) powiązania usług Web Services zawsze obsługują interakcję żądanie-odpowiedź w sposób synchroniczny. Wątek realizujący wywołanie w dostawcy usług jest blokowany do czasu otrzymania odpowiedzi z dostawcy. Więcej informacji na temat tego stylu wywołania zawiera sekcja “Wywołanie synchroniczne”.

Ważne: Dla eksportów w tym samym module nie można używać następujących kombinacji powiązań usług Web Services. Jeśli istnieje potrzeba ujawniania komponentów przy użyciu więcej niż jednego z tych powiązań eksportu, każdy z nich należy umieścić w oddzielnym module, a następnie należy połączyć te moduły z komponentami przy użyciu powiązania SCA:

- SOAP 1.1/JMS i SOAP 1.1/HTTP przy użyciu JAX-RPC,
- SOAP 1.1/HTTP przy użyciu JAX-RPC i SOAP 1.1/HTTP przy użyciu JAX-WS,
- SOAP 1.1/HTTP przy użyciu JAX-RPC i SOAP 1.2/HTTP przy użyciu JAX-WS.

Po wdrożeniu modułu SCA zawierającego powiązanie Web Service na serwerze można użyć Konsoli administracyjnej, aby wyświetlić informacje o powiązaniu lub aby zmienić wybrane właściwości powiązania.

Uwaga: Usługi Web Services umożliwiają współdziałanie aplikacji za pośrednictwem standardowych opisów usług i standardowych formatów wymienianych komunikatów. Powiązania importu i eksportu usług Web Services mogą na przykład współdziałać z usługami zaimplementowanymi przy użyciu standardów WSE (Web Services Enhancements) 3.5 i WCF (Windows Communication Foundation) 3.5 dla platformy Microsoft .NET. W przypadku współdziałania z takimi usługami użytkownik musi zagwarantować, że:

- plik WSDL (Web Services Description Language) używany w celu uzyskania dostępu do eksportu usługi Web Service zawiera niepustą wartość działania SOAP dla każdej operacji w interfejsie,
- klient usługi Web Service ustawia nagłówek SOAPAction lub nagłówek wsa:Action podczas wysyłania komunikatów do eksportu usługi Web Service.

Propagowanie nagłówków SOAP: **Advanced**

Podczas obsługi komunikatów SOAP użytkownik może wymagać dostępu do informacji z niektórych nagłówków SOAP zawartych w odbieranych komunikatach. W takim przypadku upewnij się, że komunikaty z nagłówkami SOAP są wysyłane przy użyciu konkretnych wartości lub opcji przekazywania nagłówków SOAP przez moduł.

Podczas konfigurowania powiązania usługi Web Service w produkcie Integration Designer można wskazać, że nagłówki SOAP mają być propagowane.

- Po odebraniu żądań przez eksport lub odebraniu odpowiedzi przez import istnieje możliwość uzyskania dostępu do informacji w nagłówkach SOAP. Pozwala to oprzeć logikę w module na wartościach nagłówków i umożliwia modyfikowanie tych nagłówków.
- Podczas wysyłania żądań z eksportu lub wysyłania odpowiedzi z importu istnieje możliwość dołączenia nagłówków SOAP do tych komunikatów.

Na formę i obecność propagowanych nagłówków SOAP mogą mieć wpływ zestawy strategii skonfigurowane dla importu lub eksportu. Zostało to wyjaśnione w tabeli Tabela 31 na stronie 74.

Aby skonfigurować propagowanie nagłówków SOAP dla importu lub eksportu, z widoku Właściwości produktu Integration Designer należy wybrać kartę **Propaguj nagłówek protokołu** i zaznaczyć wymagane opcje.

Nagłówek WS-Addressing

Nagłówek WS-Addressing może być propagowany przez powiązanie usługi Web Service JAX-WS.

W przypadku propagowania nagłówka WS-Addressing należy uwzględnić następujące informacje:

- Jeśli zostanie włączone propagowanie nagłówka WS-Addressing, nagłówek ten będzie propagowany do modułu w następujących okolicznościach:
 - podczas odbierania żądań przez eksport,
 - podczas odbierania odpowiedzi przez import.
- Nagłówek WS-Addressing nie jest propagowany do komunikatów wychodzących z produktu IBM Business Process Manager (czyli nagłówek nie jest propagowany podczas wysyłania żądań z importu lub wysyłania odpowiedzi z eksportu).

Nagłówek WS-Security

Nagłówek WS-Security może być propagowany zarówno przez powiązanie usługi Web Service JAX-WS, jak i powiązanie usługi Web Service JAX-RPC.

Specyfikacja WS-Security usług Web Services opisuje udoskonalenia przesyłania komunikatów SOAP, które mają na celu zwiększenie jakości zabezpieczeń za pomocą mechanizmu integralności komunikatów, mechanizmu poufności komunikatów oraz mechanizmu uwierzytelniania pojedynczych komunikatów. Te mechanizmy mogą być używane w połączeniu z wieloma modelami zabezpieczeń i technologiami szyfrowania.

W przypadku propagowania nagłówka WS-Security należy uwzględnić następujące informacje:

- Jeśli zostanie włączone propagowanie nagłówka WS-Security, nagłówek ten będzie propagowany przez moduł w następujących okolicznościach:
 - podczas odbierania żądań przez eksport,
 - podczas wysyłania żądań przez import,
 - podczas odbierania odpowiedzi przez import.
- Nagłówek *nie* będzie domyślnie propagowany podczas wysyłania odpowiedzi przez eksport. Jeśli jednak właściwość maszyny JVM **WSSECURITY.ECHO.ENABLED** zostanie ustawiona na wartość **true**, nagłówek będzie propagowany podczas wysyłania odpowiedzi przez eksport. W tym przypadku, jeśli nagłówek WS-Security w ścieżce żądania nie jest modyfikowany, nagłówki WS-Security mogą być automatycznie kopiowane z żądań do odpowiedzi.
- Dokładna forma komunikatu SOAP wysłanego z importu w przypadku żądania lub z eksportu w przypadku odpowiedzi może nie być w pełni zgodna z pierwotnym, odebrany komunikatem SOAP. Z tego powodu należy założyć, że wszystkie podpisy cyfrowe stają się niepoprawne. Jeśli w wysyłanych komunikatach wymagane jest zastosowanie podpisu cyfrowego, ten podpis należy ustawić przy użyciu odpowiedniego zestawu strategii bezpieczeństwa, a nagłówki WS-Security związane z podpisem cyfrowym i znajdujące się w odebranych komunikatach powinny zostać usunięte wewnątrz modułu.

Aby propagować nagłówek WS-Security, należy dołączyć schemat WS-Security do modułu aplikacji. Procedura dołączania schematu została opisana w sekcji “Uwzględnianie schematu WS-Security w module aplikacji”.

Sposób propagowania nagłówków

Sposób propagowania nagłówków zależy od ustawienia strategii bezpieczeństwa w powiązaniu importu lub eksportu. Zostało to przedstawione w tabeli Tabela 31:

Tabela 31. Sposób przekazywania nagłówków zabezpieczeń

	Powiązanie eksportu bez strategii bezpieczeństwa	Powiązanie eksportu ze strategią bezpieczeństwa
Powiązanie importu bez strategii bezpieczeństwa	<p>Nagłówki zabezpieczeń są przekazywane wewnątrz modułu bez zmian. Nie są one deszyfrowane.</p> <p>Nagłówki są wysyłane w tej samej formie, w której zostały odebrane.</p> <p>Podpis cyfrowy może stać się niepoprawny.</p>	<p>Nagłówki zabezpieczeń są deszyfrowane i przekazywane przez moduł z uwzględnieniem weryfikacji sygnatury i uwierzytelnienia.</p> <p>Wysyłane są deszyfrowane nagłówki.</p> <p>Podpis cyfrowy może stać się niepoprawny.</p>
Powiązanie importu ze strategią bezpieczeństwa	<p>Nagłówki zabezpieczeń są przekazywane wewnątrz modułu bez zmian. Nie są one deszyfrowane.</p> <p>Nagłówki nie powinny być propagowane do importu. W przeciwnym razie występuje błąd spowodowany duplikowaniem.</p>	<p>Nagłówki zabezpieczeń są deszyfrowane i przekazywane przez moduł z uwzględnieniem weryfikacji sygnatury i uwierzytelnienia.</p> <p>Nagłówki nie powinny być propagowane do importu. W przeciwnym razie występuje błąd spowodowany duplikowaniem.</p>

Należy skonfigurować odpowiednie zestawy strategii dla powiązań eksportu i importu, ponieważ pozwala to odizolować requester usług od zmian w konfiguracji lub wymagań QoS dostawcy usług. Uwidocznienie standardowych nagłówków SOAP w module może następnie posłużyć do wywarcia wpływu na przetwarzanie w module (na przykład rejestrowanie i śledzenie). Propagowanie nagłówków SOAP z odebranego komunikatu przez moduł zmniejsza korzyści płynące z izolowania modułu.

Standardowe nagłówki, takie jak nagłówki WS-Security, nie powinny być propagowane z żądania do importu ani z odpowiedzi do eksportu, jeśli import lub eksport mają powiązany zestaw strategii, który powoduje generowanie tych nagłówków. W przeciwnym razie wystąpi błąd spowodowany duplikowaniem nagłówków. Zamiast tego należy jawnie usunąć nagłówki albo powiązanie importu lub eksportu powinno być skonfigurowane tak, aby uniemożliwić propagowanie nagłówków protokołu.

Uzyskiwanie dostępu do nagłówków SOAP

Jeśli komunikat zawierający nagłówki SOAP zostanie odebrany z importu lub eksportu usługi Web Service, nagłówki są umieszczane w sekcji nagłówków obiektu SMO. Użytkownik może uzyskać dostęp do informacji w nagłówku. Zostało to opisane w sekcji “Uzyskiwanie dostępu do informacji w nagłówku SOAP w obiekcie SMO”.

Uwzględnianie schematu WS-Security w module aplikacji

Poniższa procedura opisuje kroki związane z dołączaniem schematu do modułu aplikacji:

- Jeśli komputer, na którym działa produkt Integration Designer, jest podłączony do Internetu, wykonaj następujące kroki:
 1. W perspektywie Integracja biznesowa wybierz opcję **Zależności** dla projektu.
 2. Rozwiń sekcję **Zasoby predefiniowane** i wybierz opcję **Pliki schematu WS-Security 1.0** lub **Pliki schematu WS-Security 1.1**, aby zaimportować schemat do modułu.

3. Oczyszczyć i odbudować projekt.
- Jeśli komputer, na którym działa produkt Integration Designer, nie jest podłączony do Internetu, można pobrać schemat na drugi komputer, który jest podłączony do Internetu. Można następnie skopiować te pliki na komputer, na którym działa produkt Integration Designer.
 1. Z poziomu komputera podłączonego do Internetu pobrać schemat zdalny:
 - a. Kliknij opcję **Plik > Importuj > Integracja biznesowa > WSDL i XSD**.
 - b. Wybierz opcję **Zdalny plik WSDL** lub **Plik XSD**.
 - c. Zimportuj następujące schematy:
 - `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. Skopiuj schematy na komputer, który nie jest podłączony do Internetu.
 3. Zimportuj schemat na komputerze, który nie jest podłączony do Internetu:
 - a. Kliknij opcję **Plik > Importuj > Integracja biznesowa > WSDL i XSD**.
 - b. Wybierz opcję **Lokalny plik WSDL** lub **Plik XSD**.
 4. Zmień położenia schematów dla pliku `oasis-wss-wssecurity-secext-1.1.xsd`:
 - a. Otwórz schemat znajdujący się w następującym miejscu: *położenie_robocze/nazwa_modułu* `StandardImportFilesGen/oasis-wss-wssecurity-secext-1.1.xsd`.
 - b. Zmień element:


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

 na element:


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


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

 na element:


```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
              schemaLocation='../w3/tr/_2002/rec_xmlenc_core_20021210/xenc-schema.xsd' />
```
 5. Zmień położenie schematu dla pliku `oasis-200401-wss-wssecurity-secext-1.0.xsd`:
 - a. Otwórz schemat znajdujący się w następującym miejscu: *położenie_robocze/nazwa_modułu* `StandardImportFilesGen/oasis-200401-wss-wssecurity-secext-1.0.xsd`.
 - b. Zmień element:


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

 na element:


```
<xsd:import namespace="http://www.w3.org/2000/09/xmlsig#"
              schemaLocation='../w3/tr/_2002/rec_xmlsig_core_20020212/xmlsig-core-schema.xsd" />
```
6. Oczyszczyć i odbudować projekt.

Propagowanie nagłówków transportu: **Advanced**

Podczas obsługi komunikatów SOAP użytkownik może wymagać dostępu do informacji z niektórych nagłówków transportu zawartych w odbieranych komunikatach. W takim przypadku upewnij się, że komunikaty z nagłówkami SOAP są wysyłane przy użyciu konkretnych wartości lub opcji przekazywania nagłówków SOAP przez moduł.

Podczas konfigurowania powiązania usługi Web Service w produkcie Integration Designer można wskazać, że nagłówki transportu mają być propagowane.

- Po odebraniu żądań przez eksport lub odebraniu odpowiedzi przez import istnieje możliwość uzyskania dostępu do informacji w nagłówkach transportu. Pozwala to oprzeć logikę w module na wartościach nagłówków i umożliwia modyfikowanie tych nagłówków.
- Podczas wysyłania odpowiedzi z eksportu lub wysyłania żądań z importu istnieje możliwość dołączenia nagłówków transportu do tych komunikatów.

Określanie propagowania nagłówków

Aby skonfigurować propagowanie nagłówków transportu dla importu lub eksportu, wykonaj następujące kroki:

1. W widoku Właściwości produktu Integration Designer wybierz opcję **Powiązanie > Propagacja**.
2. Ustaw wymagane opcje propagowania nagłówka transportu.

Uwaga: Propagowanie nagłówków transportu jest domyślnie wyłączone i może być wdrożone tylko w środowisku wykonawczym w wersji 7.0.0.3 (lub późniejszej wersji). Należy również zauważyć, że w przypadku wersji 7.0.0.3 propagowanie nagłówków transportu jest możliwe tylko w przypadku nagłówków transportu HTTP.

Jeśli zostanie włączone propagowanie nagłówków transportu, nagłówki będą propagowane z komunikatów odebranych przez moduł i jeśli nie zostaną one jawnie usunięte, będą używane w kolejnych wywołaniach w tym samym wątku.

Uwaga: Nagłówków transportu nie można propagować, jeśli używane jest powiązanie usługi Web Service JAX-RPC.

Uzyskiwanie dostępu do informacji w nagłówku

Jeśli włączone jest propagowanie nagłówków transportu dla odbieranych komunikatów, wszystkie nagłówki transportu (w tym nagłówki niestandardowe) są widoczne w obiekcie komunikatu usługi (SMO). Nagłówkom można nadać inne wartości, ale można również utworzyć nowe nagłówki. Należy jednak zauważyć, że nie jest przeprowadzane żadne sprawdzanie ani sprawdzanie poprawności ustawianych wartości. Niepoprawne lub nieodpowiednie nagłówki mogą zaś powodować problemy z usługą Web Service w czasie wykonywania.

Podczas ustawiania nagłówków HTTP należy uwzględnić następujące informacje:

- Wszystkie zmiany nagłówków, które są zastrzeżone dla mechanizmu usługi Web Service, nie będą uwzględniane w komunikacie wychodzącym. Nagłówkami zastrzeżonymi dla mechanizmu usługi Web Service są na przykład: wersja HTTP, metoda HTTP, nagłówek Content-Type, nagłówek Content-Length i nagłówek SOAPAction.
- Jeśli wartość nagłówka jest liczbą, należy ustawić bezpośrednio liczbę (a nie łańcuch). Na przykład należy użyć definicji **Max-Forwards = 5** (a nie **Max-Forwards = Max-Forwards: 5**) lub **Age = 300** (a nie **Age = Age: 300**).
- Jeśli komunikat żądania jest mniejszy niż 32 kB, mechanizm usługi Web Service usuwa nagłówek Transfer-Encoding i ustawia nagłówek Content-Length na ustaloną wielkość komunikatu.
- Nagłówek Content-language jest resetowany przez mechanizm WAS.channel.http w ścieżce odpowiedzi.
- Niepoprawne ustawienie nagłówka Upgrade powoduje błąd 500.
- Następujące nagłówki dopisują wartość zastrzeżoną przez mechanizm usługi Web Service do ustawień klienta:
 - User-Agent,
 - Cache-Control,
 - Pragma,
 - Accept,
 - Connection.

Dostęp do informacji w nagłówku można uzyskać na jeden z następujących sposobów:

- Korzystając z operacji podstawowej mediacji w celu uzyskania dostępu do struktur obiektu SMO
Informacje na temat używania operacji podstawowych mediacji zawiera sekcja z odsyłaczami “Informacje pokrewne”.
- Korzystając z interfejsu SPI usługi kontekstu

Poniższy przykładowy kod odczytuje nagłówki transportu HTTP z usługi kontekstu:

```
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();
    }
}
```

Rozwiązywanie problemów

Jeśli podczas wysyłania zmienionych nagłówków wystąpią problemy, można przechwycić komunikat TCP/IP za pomocą takich narzędzi jak monitor TCP/IP w produkcie Integration Designer. Aby uzyskać dostęp do monitora TCP/IP, należy wybrać opcję **Uruchom/debuguj > Monitor TCP/IP** na stronie Preferencje.

Wartości nagłówków można również wyświetlić przy użyciu śledzenia mechanizmu JAX-WS: **org.apache.axis2.*=all: com.ibm.ws.websvcs.*=all:**

Praca z powiązaniem usługi Web Services JAX-WS: **Advanced**

Jeśli w aplikacjach używane są powiązania usługi Web Service JAX-WS, istnieje możliwość dodania jakości usług (QoS) SAML (Security Assertion Markup Language) do powiązania. W tym celu należy najpierw użyć Konsoli administracyjnej do zaimportowania zestawu strategii. Konsoli administracyjnej można również użyć, aby upewnić się, czy serwer jest poprawnie skonfigurowany pod kątem użycia wraz z powiązaniem usługi Web Service JAX-WS.

Importowanie zestawów strategii SAML: **Advanced**

Język Security Assertion Markup Language (SAML) jest opartym na formacie XML standardem OASIS umożliwiającym wymianianie informacji o tożsamości użytkownika i atrybutach zabezpieczeń. Zestaw strategii SAML można określić, konfigurując powiązanie usługi Web Service JAX-WS w produkcie Integration Designer. Najpierw za pomocą Konsoli administracyjnej produktu IBM Business Process Manager należy udostępnić zestawy strategii SAML, aby można je było zaimportować do produktu Integration Designer.

Zestawy strategii SAML zwykle znajdują się w katalogu konfiguracji profilu:

katalog_główny_profilu/config/templates/PolicySets

Przed rozpoczęciem wykonywania tej procedury należy sprawdzić, czy następujące katalogi (zawierające zestawy strategii) znajdują się w katalogu konfiguracji profilu:

- SAML11 Bearer WSHTTPS default
- SAML20 Bearer WSHTTPS default
- SAML11 Bearer WSSecurity default
- SAML20 Bearer WSSecurity default
- SAML11 HoK Public WSSecurity default
- SAML20 HoK Public WSSecurity default
- SAML11 HoK Symmetric WSSecurity default
- SAML20 HoK Symmetric WSSecurity default

- Username WSHTTPS default

Jeśli katalogów tych nie ma w katalogu konfiguracji profilu, należy je tam skopiować z następującego położenia:

katalog_główny_serwera_aplikacji/profileTemplates/default/documents/config/templates/PolicySets

Importując zestawy strategii do Konsoli administracyjnej, należy wybrać te z nich, które mają być dostępne dla produktu Integration Designer, a następnie zapisać plik .zip dla każdego z tych zestawów strategii w miejscu dostępnym dla produktu Integration Designer.

1. Zaimportuj zestawy strategii, wykonując następujące kroki:
 - a. W Konsoli administracyjnej wybierz opcję **Usługi > Zestawy strategii > Zestawy strategii aplikacji**.
 - b. Kliknij opcję **Importuj > Z repozytorium domyślnego**.
 - c. Wybierz domyślne zestawy strategii SAML i kliknij przycisk **OK**.
2. Wyeksportuj zestawy strategii, aby można było ich użyć w produkcie Integration Designer:
 - a. Na stronie zestawów strategii aplikacji wybierz zestaw strategii SAML, który należy wyeksportować, i kliknij opcję **Eksportuj**.

Uwaga: Jeśli strona zestawów strategii aplikacji nie jest aktualnie wyświetlana, kliknij opcję **Usługi > Zestawy strategii > Zestawy strategii aplikacji** w Konsoli administracyjnej.

- b. Na następnej stronie kliknij odsyłacz do pliku .zip zestawu strategii.
- c. W oknie Pobieranie pliku kliknij przycisk **Zapisz** i wskaż miejsce dostępne dla produktu Integration Designer.
- d. Kliknij przycisk **Wstecz**.
- e. Wykonaj kroki od 2a do 2d dla każdego eksportowanego zestawu strategii.

Zestawy strategii SAML zostaną zapisane w plikach .zip i będą gotowe do zaimportowania do produktu Integration Designer.

Zestawy strategii należy zaimportować do produktu Integration Designer w sposób opisany w temacie “Zestawy strategii”.

Wywoływanie usług Web Services wymagających podstawowego uwierzytelniania HTTP: 

W podstawowym uwierzytelnianiu HTTP do uwierzytelnienia klienta usługi wobec zabezpieczonego punktu końcowego stosowane są nazwa użytkownika oraz hasło. Podstawowe uwierzytelnianie HTTP można skonfigurować na potrzeby wysyłania i odbierania żądań usług Web Services.

Skonfigurowanie podstawowego uwierzytelniania HTTP na potrzeby odbierania żądań usługi Web Service polega na skonfigurowaniu interfejsu API języka Java API dla powiązania elementu eksportowanego XML usług Web Services JAX-WS w sposób opisany w temacie Tworzenie i przypisywanie ról zabezpieczeń do elementów eksportowanych usług Web Service.

Podstawowe uwierzytelnianie HTTP można włączyć na potrzeby żądań usług Web Services, które są wysyłane przez powiązanie elementu importowanego JAX-WS, w jeden z dwóch sposobów:

- Podczas konfigurowania powiązania elementu importowanego w module SCA można wybrać dostarczony zestaw strategii uwierzytelniania HTTP o nazwie BPMHTTPBasicAuthentication (udostępniany wraz z powiązaniem elementu importowanego usługi Web Service JAX-WS) lub dowolny inny zestaw strategii obejmujący strategię HTTPTransport.
- Podczas konstruowania modułu SCA można użyć możliwości przepływu mediacji do dynamicznego utworzenia nowego nagłówka uwierzytelniania HTTP oraz określenia w tym nagłówku informacji o nazwie użytkownika i hasle.

Uwaga: Zestaw strategii ma pierwszeństwo przed wartością określoną w nagłówku. Aby w czasie wykonywania użyć wartości ustawionej w nagłówku uwierzytelniania HTTP, nie należy załączać zestawu strategii obejmującego strategię

HTTPTransport. W szczególności nie należy używać domyślnego zestawu strategii BPMHTTPBasicAuthentication oraz, jeśli zdefiniowano zestaw strategii, należy upewnić się, że nie obejmuje on strategii HTTPTransport.

Więcej informacji na temat zestawów strategii i powiązań strategii usług Web Services, jak również sposobu ich używania można znaleźć w temacie Web services policy sets (Zestawy strategii usług Web Services) w Centrum informacyjnym serwera WebSphere Application Server.

- Aby użyć dostarczonego zestawu strategii, wykonaj następujące kroki:
 1. Opcjonalne: W Konsoli administracyjnej utwórz powiązanie strategii ogólnej klienta obejmującej strategię HTTPTransport z wartościami wymaganego identyfikatora użytkownika i hasła lub przeprowadź edycję istniejącego.
 2. W produkcie IBM Integration Designer wygeneruj powiązanie elementu importowanego usługi Web Service JAX-WS i przyłącz do niego zestaw strategii BPMHTTPBasicAuthentication.
 3. Wykonaj *jeden* z następujących kroków:
 - W produkcie IBM Integration Designer we właściwościach powiązania elementu importowanego usługi Web Service JAX-WS określ nazwę istniejącego powiązania strategii ogólnej klienta obejmującej strategię HTTPTransport.
 - Po wdrożeniu modułu SCA użyj Konsoli administracyjnej, aby wybrać istniejące powiązanie strategii klienta lub utworzyć nowe i powiązać je z powiązaniem importu.
 4. Opcjonalne: W Konsoli administracyjnej serwera Process Server przeprowadź edycję wybranego powiązania zestawu strategii, określając wymagany identyfikator i hasło.
- Aby określić nazwę użytkownika i hasło w nagłówku uwierzytelniania HTTP, wykonaj jeden z następujących zestawów kroków:
 - Użyj w produkcie IBM Integration Designer operacji podstawowej mediacji procedury ustawiającej nagłówek HTTP do utworzenia nagłówka uwierzytelniania HTTP i określ nazwę użytkownika oraz hasło.
 - Jeśli wymagana jest dodatkowa logika, użyj kodu w języku Java w niestandardowej operacji podstawowej mediacji (w sposób przedstawiony w poniższym przykładzie) realizującej następujące działania:
 1. Utworzenie nagłówka uwierzytelniania HTTP.
 2. Określenie nazwy użytkownika i hasła.
 3. Dodanie nowego nagłówka uwierzytelniania HTTP do elementu sterującego HTTPControl.
 4. Ustawienie zaktualizowanego elementu sterującego HTTPControl z powrotem w usłudze Kontekst.

```
// Pobranie typu HeaderInfoType z usługi kontekstu
ContextService contextService = (ContextService) ServiceManager.INSTANCE
    .locateService("com/ibm/bpm/context/ContextService");
HeaderInfoType headers = contextService.getHeaderInfo();
if(headers == null){
    headers = ContextObjectFactory.eINSTANCE.createHeaderInfoType();
}
// Pobranie nagłówka HTTP i elementu sterującego HTTPControl z typu HeaderInfoType
HTTPHeaderType httpHeaderType = headers.getHTTPHeader();
HTTPControl cp = httpHeaderType.getControl();
HeadersFactory factory = HeadersFactory.eINSTANCE;
if(cp == null){
    cp = factory.createHTTPControl();
}
// Utworzenie nowego obiektu HTTPAuthentication i ustawienie referencji 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);
// Ustawienie informacji nagłówka z powrotem na bieżący kontekst wykonywania.
contextService.setHeaderInfo(headers);
```

Sprawdzanie konfiguracji serwera: **Advanced**

Wdrażając aplikację z powiązaniem z usługą Web Service JAX-WS, należy upewnić się, że serwer, na którym ta aplikacja jest wdrożona, nie ma zaznaczonej opcji **Start components as needed** (Uruchamiaj komponenty, gdy są potrzebne).

Aby sprawdzić, czy opcja ta jest zaznaczona, wykonaj następujące kroki w Konsoli administracyjnej:

1. Kliknij opcję **Serwery > Typy serwerów > Serwery aplikacji WebSphere**.
2. Kliknij nazwę serwera.
3. Na karcie Konfiguracja sprawdź, czy opcja **Start components as needed** (Uruchamiaj komponenty, gdy są potrzebne) jest zaznaczona.
4. Wykonaj jeden z następujących kroków:
 - Jeśli opcja **Start components as needed** (Uruchamiaj komponenty, gdy są potrzebne) jest zaznaczona, usuń zaznaczenie i kliknij przycisk **Zastosuj**.
 - Jeśli opcja **Start components as needed** (Uruchamiaj komponenty, gdy są potrzebne) nie jest zaznaczona, kliknij przycisk **Anuluj**.

Załączniki w komunikatach SOAP: **Advanced**

Istnieje możliwość wysyłania i odbierania komunikatów SOAP, które zawierają dane binarne (na przykład pliki PDF lub obrazy JPEG) dołączone w formie załączników. Załączniki mogą być *przywoływane* (czyli reprezentowane jawnie jako części komunikatu w interfejsie usługi) lub *nieprzywoływane* (w tym przypadku możliwe jest dołączenia dowolnej liczby i dowolnego typu załączników).

Przywoływany załącznik może być reprezentowany w jeden z następujących sposobów:

- W załącznikach MTOM stosowane jest kodowanie używane w mechanizmie SOAP Message Transmission Optimization Mechanism (<http://www.w3.org/TR/soap12-mtom/>). Możliwość korzystania z załączników MTOM włącza się za pomocą odpowiedniej opcji konfiguracyjnej w powiązaniach importu i eksportu. Jest to zalecana metoda kodowania załączników dla nowych aplikacji.
- Jako element wsi:swaRef-typed w schemacie komunikatu
Załączniki zdefiniowane przy użyciu typu wsi:swaRef są zgodne ze standardem organizacji WS-I (Web Services Interoperability Organization) *Attachments Profile Version 1.0* (profil załącznika, wersja 1.0 - <http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), który definiuje relacje między elementami komunikatu i częściami MIME.
- Jako część komunikatu najwyższego poziomu, przy użyciu typu schematu binarnego
Załączniki reprezentowane jako części komunikatu najwyższego poziomu są zgodne ze specyfikacją *SOAP Messages with Attachments* (komunikaty SOAP z załącznikami - <http://www.w3.org/TR/SOAP-attachments>).
Załączniki reprezentowane jako części komunikatu najwyższego poziomu mogą również być skonfigurowane tak, aby zagwarantować że dokument WSDL i komunikaty utworzone przez powiązanie są zgodne ze standardami WS-I *Attachments Profile Version 1.0* (profil załącznika, wersja 1.0) i WS-I *Basic Profile Version 1.1* (profil podstawowy, wersja 1.1 - <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>).

Nieprzywoływany załącznik jest przenoszony w komunikacie SOAP bez żadnej reprezentacji w schemacie komunikatu.

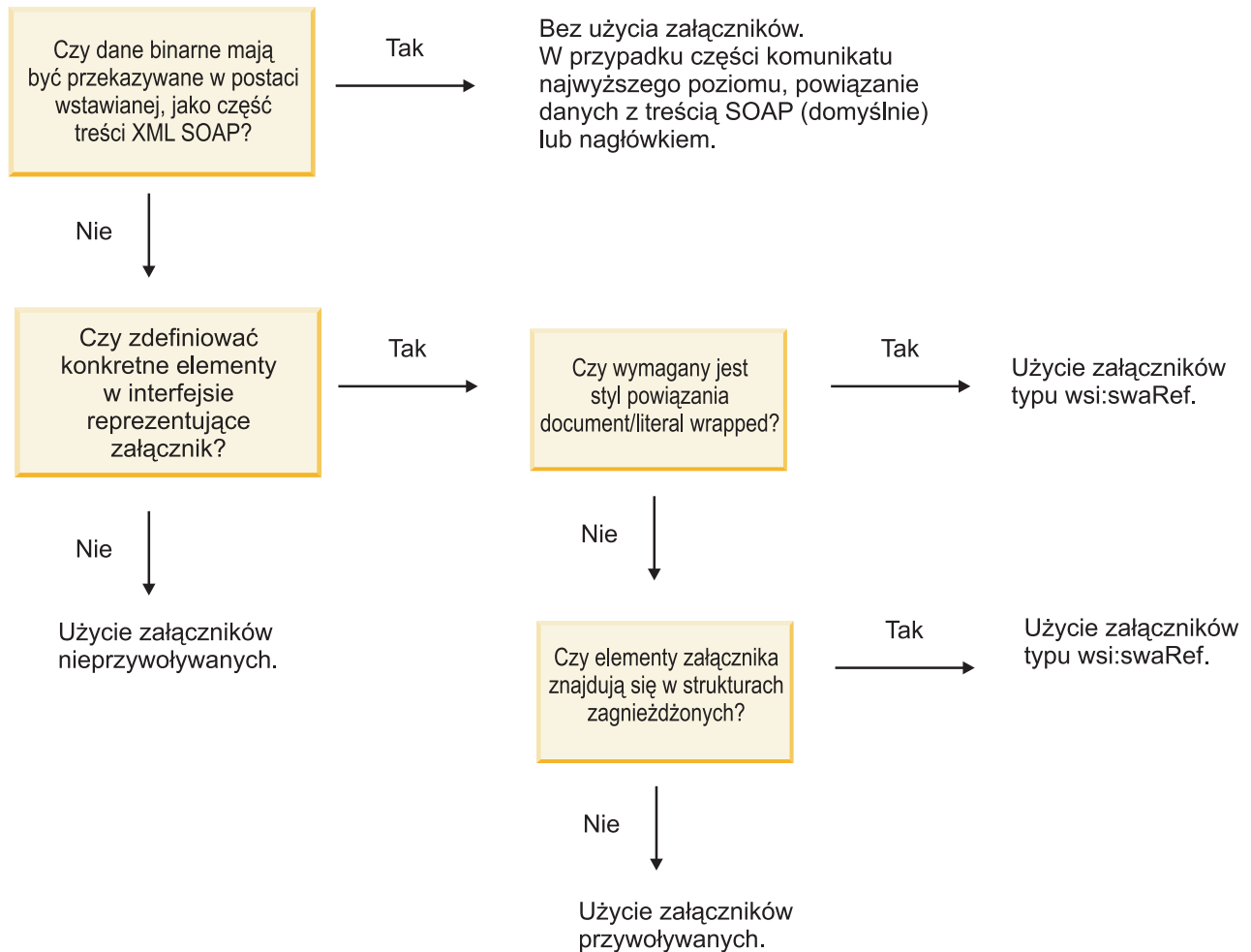
We wszystkich przypadkach, z wyjątkiem załączników MTOM, powiązanie SOAP WSDL powinno zawierać powiązanie MIME dla załączników, które mają zostać użyte, a maksymalna wielkość załączników nie powinna przekraczać 20 MB.

Uwaga: Aby wysyłać lub odbierać komunikaty SOAP z załącznikami, należy używać jednego z powiązań usług Web Services opartego na specyfikacji JAX-WS (Java API for XML Web Services).

Jak wybrać odpowiedni styl załącznika: **Advanced**

Podczas projektowania nowego interfejsu usługi, który zawiera dane binarne, należy uwzględnić sposób przenoszenia tych danych binarnych w komunikatach SOAP wysyłanych i odbieranych przez usługę.

Mechanizm Message Transmission Optimization Mechanism (MTOM) należy stosować na potrzeby załączników, gdy obsługuje go podłączona aplikacja usługi Web Service. Jeśli tak nie jest, należy wybrać inny styl załączników. Jest to przedstawione na poniższym diagramie. Przy określaniu odpowiedniego stylu załącznika należy posłużyć się następującymi pytaniami:



Załączniki MTOM: części komunikatu najwyższego poziomu:

Istnieje możliwość wysyłania i odbierania komunikatów usług Web Services zawierających załączniki MTOM (Message Transmission Optimization Mechanism) SOAP. W komunikacie SOAP z wieloma częściami MIME pierwszą częścią komunikatu jest treść SOAP, a załącznik lub załączniki znajdują się w kolejnych częściach.

Przy wysyłaniu lub odbieraniu przywoływanego załącznika w komunikacie SOAP dane binarne, które tworzą załącznik (często dość duży), są przechowywane niezależnie od treści komunikatu SOAP, a więc nie muszą być analizowane jako kod XML. Powoduje to, że przetwarzanie jest wydajniejsze niż byłoby, gdyby dane binarne były przechowywane w elemencie XML.

Poniżej przedstawiony jest przykładowy komunikat MTOM SOAP:


```

... inne nagłówki transportowe ...
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>

... w tym miejscu są dane binarne ...
--MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812--

```

Należy zwrócić uwagę, że w tym przykładzie mechanizmu MTOM typem treści koperty SOAP jest **application/xop+xml**, a dane binarne są zastąpione następującym elementem **xop:Include**:

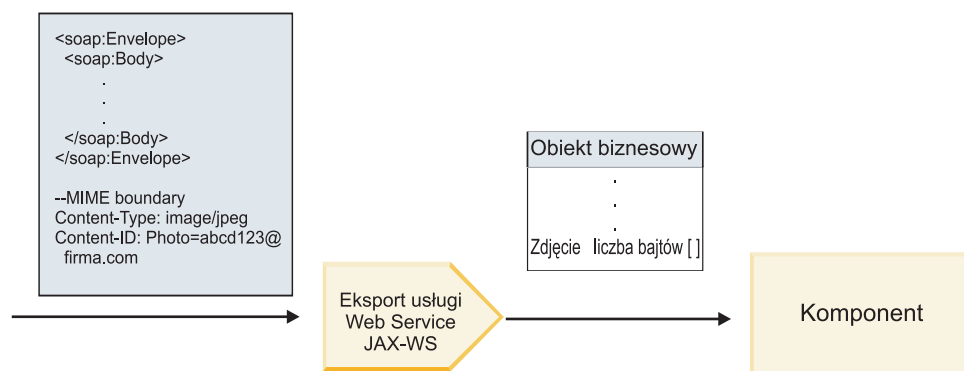
```

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

```

Przetwarzanie danych przychodzących załączników przywoływanych

Jeśli klient przekaże komunikat SOAP z załącznikiem do komponentu SCA (Service Component Architecture), powiązanie eksportu usługi Web Service JAX-WS najpierw usuwa załącznik. Następnie analizuje część SOAP komunikatu i tworzy obiekt biznesowy. Na koniec, powiązanie umieszcza binarny załącznik w obiekcie biznesowym.



Rysunek 13. Sposób przetwarzania przez powiązanie eksportu usługi Web Service (JAX-WS) komunikatu SOAP z załącznikiem przywoływanym

Atrybuty załączników MTOM

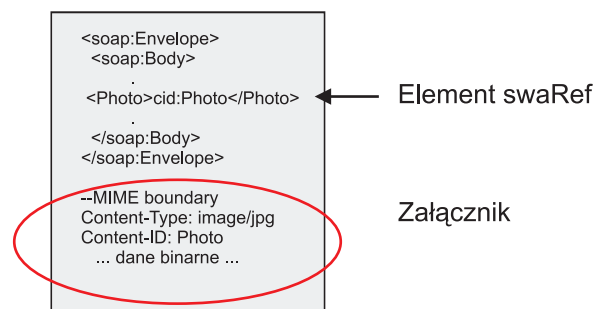
- Mechanizm MTOM może obsługiwać elementy załącznika w obrębie struktur zagnieżdżonych.
- Mechanizm MTOM jest dostępny jedynie dla typu base64Binary.
- Mechanizm MTOM może obsługiwać elementy załącznika w obrębie struktur zagnieżdżonych. Oznacza to, że elementy **bodyPath** związane z załącznikami MTOM określają położenie **xpath** elementu, w którym przechowywany jest załącznik MTOM. Logika obliczania wartości elementu **bodyPath** ściśle odpowiada schematowi generowania położenia **xpath** przedstawionemu w następujących przykładach:
 - W przypadku typu innego niż tablicowy (**maxOccurs** jest równe 1): /sendImage/input/imageData
 - W przypadku typu tablicowego (**maxOccurs** > 1): /sendImage/input/imageData[1]
- Mieszane typy załączników nie są obsługiwane. Oznacza to, że jeśli mechanizm MTOM jest włączony w powiązaniu importu, generowany będzie załącznik MTOM. Jeśli mechanizm MTOM jest wyłączony lub jeśli pozostawiono domyślną wartość konfiguracji mechanizmu MTOM w powiązaniu eksportu, przychodzący komunikat MTOM nie jest obsługiwany.

Załączniki przywoływane: elementy typu swaRef: **Advanced**

Istnieje możliwość wysyłania i odbierania komunikatów SOAP zawierających załączniki reprezentowane w interfejsie usługi jako elementy typu swaRef.

Element typu swaRef jest zdefiniowany w standardzie organizacji WS-I (Web Services Interoperability Organization) *Attachments Profile* Version 1.0 (profil załącznika, wersja 1.0 - <http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), który definiuje relacje między elementami komunikatu i częściami MIME.

Treść komunikatu SOAP zawiera element typu swaRef określający identyfikator treści załącznika.



Rysunek 14. Komunikat SOAP z elementem swaRef

Dokument WSDL dla tego komunikatu SOAP zawiera element swaRef w części komunikatu identyfikującej załącznik.

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

Dokument WSDL powinien też zawierać powiązanie MIME wskazujące, że mają być używane wieloczęściowe komunikaty MIME.

Uwaga: Dokument WSDL *nie* zawiera powiązania MIME dla konkretnego elementu komunikatu typu swaRef, ponieważ powiązania MIME dotyczą tylko części komunikatu najwyższego poziomu.

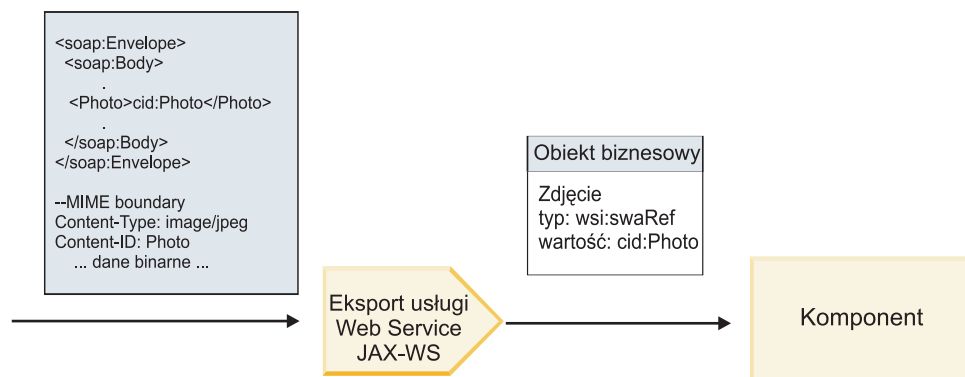
Załączniki reprezentowane jako elementy typu swaRef mogą być propagowane tylko przez komponenty przepływu mediacji. Jeśli załącznik wymaga dostępu z poziomu innego typu komponentu lub propagowania do innego typu komponentu, należy użyć komponentu przepływu mediacji w celu przeniesienia załącznika do miejsca, które jest dostępne dla tego komponentu.

Przetwarzanie danych przychodzących załączników

Do konfigurowania powiązania eksportu w celu odebrania załącznika używany jest produkt Integration Designer. Użytkownik tworzy moduł i powiązany interfejs oraz operacje, między innymi element typu swaRef. Następnie tworzy powiązanie usługi Web Service JAX-WS.

Uwaga: Szczegółowe informacje zawiera temat “Praca z załącznikami” w Centrum informacyjnym produktu Integration Designer.

Jeśli klient przekaże komunikat SOAP z załącznikiem swaRef do komponentu SCA (Service Component Architecture), powiązanie eksportu usługi Web Service JAX-WS najpierw usuwa załącznik. Następnie analizuje część SOAP komunikatu i tworzy obiekt biznesowy. Na koniec, powiązanie ustawia identyfikator treści załącznika w obiekcie biznesowym.



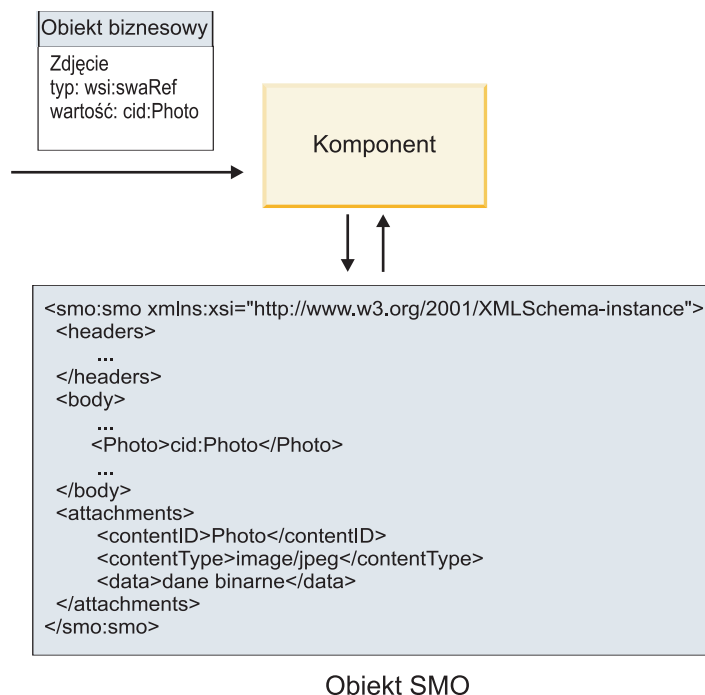
Rysunek 15. Sposób przetwarzania komunikatu SOAP zawierającego załącznik swaRef przez powiązanie eksportu usługi Web Service JAX-WS

Uzyskiwanie dostępu do metadanych załącznika w komponencie przepływu mediacji

Jak przedstawiono na ilustracji Rys. 16 na stronie 85, kiedy komponenty uzyskują dostęp do załączników swaRef, identyfikator treści załącznika jest wyświetlany jako element typu swaRef.

Dla każdego załącznika komunikatu SOAP istnieje odpowiadający mu element **attachments** w obiekcie SMO. Jeśli używany jest typ swaRef profilu WS-I, element **attachments** zawiera typ treści załącznika oraz identyfikator treści, a także rzeczywiste dane binarne załącznika.

Aby uzyskać wartość załącznika swaRef, konieczne jest więc uzyskanie wartości elementu typu swaRef, a następnie znalezienie elementu **attachments** z odpowiednią wartością **contentID**. Należy zauważyć, że w przypadku wartości **contentID** przedrostek **cid:** jest zazwyczaj usuwany z wartości swaRef.



Rysunek 16. Sposób wyświetlania załączników swaRef w obiekcie SMO

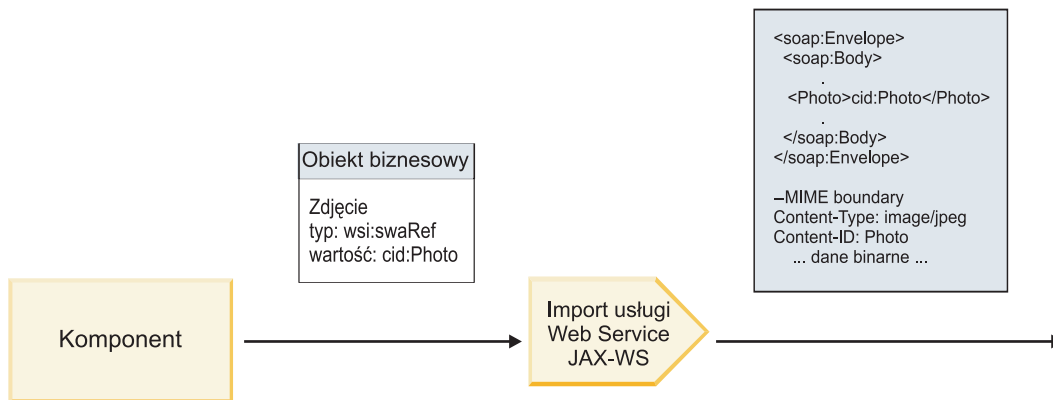
Przetwarzanie danych wychodzących

Do skonfigurowania powiązania importu usługi Web Service JAX-WS w celu wywołania zewnętrznej usługi Web Service używany jest produkt Integration Designer. Powiązanie importu jest konfigurowane przy użyciu dokumentu WSDL, który opisuje usługę Web Service do wywołania oraz definiuje załącznik, który zostanie przekazany do usługi Web Service.

Po otrzymaniu komunikatu SCA przez powiązanie importu usługi Web Service JAX-WS, elementy typu swaRef są wysyłane jako załączniki, jeśli import jest połączony z komponentem przepływu mediacji, a element typu swaRef ma odpowiedni element **attachments**.

W przypadku przetwarzania danych wychodzących elementy typu swaRef są zawsze wysyłane wraz z ich wartościami identyfikatora treści. Jednak moduł mediacji musi gwarantować, że istnieje odpowiedni element **attachments** ze zgodną wartością **contentID**.

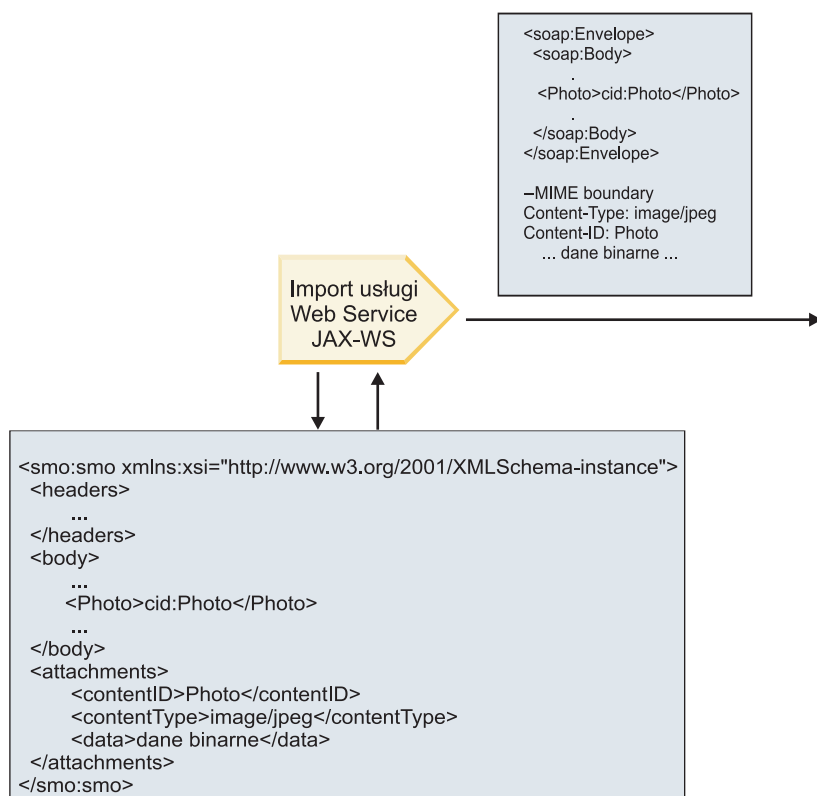
Uwaga: Aby zachować zgodność z profilem załączników WS-I, wartość **content ID** powinna być zgodna z kodowaniem części identyfikatora content-id opisanym w sekcji 3.8 dokumentu WS-I *Attachments Profile* 1.0.



Rysunek 17. Sposób generowania komunikatu SOAP zawierającego załącznik swaRef przez powiązanie importu usługi Web Service JAX-WS

Ustawianie metadanych załącznika w komponencie przepływu mediacji

Jeśli w obiekcie SMO występuje wartość elementu typu swaRef oraz element **attachments**, powiązanie przygotowuje komunikat SOAP (wraz z załącznikiem) i wysyła ten komunikat do odbiorcy.



Obiekt SMO

Rysunek 18. Sposób dostępu do załącznika swaRef w obiekcie SMO w celu utworzenia komunikatu SOAP

Obiekt SMO zawiera element **attachments** tylko jeśli komponent przepływu mediacji jest połączony bezpośrednio z importem lub eksportem. Element ten nie jest przekazywany przez inne typy komponentów. Jeśli potrzebne są wartości znajdujące się w module zawierającym inne typy komponentów, należy użyć komponentu przepływu mediacji w celu skopiowania tych wartości do miejsca, w którym będzie możliwe uzyskiwanie do nich dostępu z poziomu modułu.

Należy również użyć innego komponentu przepływu mediacji do ustawienia poprawnych wartości przed zrealizowaniem wywołania wychodzącego za pośrednictwem importu usługi Web Service.

Ważne: Operacja podstawowa mediacji Odzworowanie transformuje komunikaty przy użyciu transformacji XSLT 1.0. Zostało to opisane w sekcji “Reprezentacja XML obiektu SMO”. Transformacja opiera się na serializacji XML obiektu SMO. Operacja podstawowa mediacji Odzworowanie umożliwia określenie elementu głównej serializacji, a element główny dokumentu XML odzwierciedla ten element głównej serializacji.

Podczas wysyłania komunikatów SOAP z załącznikami wybrany element główny określa sposób propagowania załączników.

- Jeśli jako element główny odzworowania XML zostanie użyty element “/body”, wszystkie załączniki są domyślnie propagowane do całego odzworowania.
- Jeśli jako element główny odzworowania zostanie użyty element “/”, można kontrolować propagowanie załączników.

Załączniki przywoływane: części komunikatu najwyższego poziomu: 

Istnieje możliwość wysyłania i odbierania komunikatów SOAP zawierających załączniki binarne, które są zadeklarowane jako części w interfejsie usługi.

W komunikacie SOAP z wieloma częściami MIME pierwszą częścią komunikatu jest treść SOAP, a załącznik lub załączniki znajdują się w kolejnych częściach.

Jaka korzyść płynie z wysyłania lub odbierania przywoływanego załącznika w komunikacie SOAP? Dane binarne, które tworzą załącznik (często dość duży) są przechowywane niezależnie od treści komunikatu SOAP, a więc nie muszą być analizowane jako kod XML. Powoduje to, że przetwarzanie jest wydajniejsze niż byłoby, gdyby dane binarne były przechowywane w elemencie XML.

Typy komunikatów SOAP z załącznikami przywoływanymi

Od wersji 7.0.0.3 produktu IBM Business Process Manager użytkownik ma wybór sposobu generowania komunikatu SOAP:

- **Komunikaty zgodne ze standardem WS-I**

Środowisko wykonawcze może generować komunikaty SOAP, które są zgodne ze standardem WS-I *Attachments Profile Version 1.0* (profil załącznika, wersja 1.0) i standardem WS-I *Basic Profile Version 1.1* (profil podstawowy, wersja 1.1). W komunikacie SOAP zgodnym z tymi standardami tylko jedna część komunikatu jest powiązana z treścią komunikatu SOAP. W przypadku tych części, które są powiązane jako załączniki, do powiązania załącznika z częścią komunikatu używane jest kodowanie części identyfikatora content-id (opisane w standardzie WS-I *Attachments Profile Version 1.0*).

- **Komunikaty niezgodne ze standardami WS-I**

Środowisko wykonawcze może generować komunikaty SOAP, które nie są zgodne ze standardami WS-I, ale są zgodne z komunikatami generowanymi w wersji 7.0 lub 7.0.0.2 produktu IBM Business Process Manager. Komunikaty SOAP używają elementów najwyższego poziomu o nazwach utworzonych na podstawie części komunikatu, które zawierają atrybut **href** przechowujący identyfikator **content-id** załącznika, ale nie jest używane kodowanie części identyfikatora content-id (opisane w standardzie WS-I *Attachments Profile Version 1.0*).

Wybór zgodności ze standardami WS-I w przypadku eksportów usług Web Services

Do konfigurowania powiązania eksportu używany jest produkt Integration Designer. Użytkownik tworzy moduł i powiązany interfejs oraz operacje. Następnie tworzy powiązanie usługi Web Service JAX-WS. Strona Załączniki przywoływane wyświetla wszystkie binarne części z utworzonej operacji, a użytkownik wybiera części, które będą załącznikami. Następnie na stronie Określanie zgodności ze standardem WS-I AP 1.0 produktu Integration Designer określana jest jedna z następujących opcji:

- **Użyj komunikatu SOAP zgodnego ze standardem WS-I AP 1.0**

Jeśli zostanie wybrana ta opcja, użytkownik określa również która część komunikatu powinna być powiązana z treścią komunikatu SOAP.

Uwaga: Tej opcji można użyć tylko w sytuacji, kiedy odpowiedni plik WSDL jest również zgodny ze standardem WS-I.

Plik WSDL wygenerowany przez produkt Integration Designer 7.0.0.3 jest zgodny ze standardem WS-I. Jeśli jednak zostanie zaimportowany plik WSDL, który nie jest zgodny ze standardem WS-I, nie można wybrać tej opcji.

- **Użyj komunikatu SOAP niezgodnego ze standardem WS-I AP 1.0**

Jeśli zostanie wybrana ta opcja (która jest opcją domyślną), z treścią komunikatu SOAP powiązana jest pierwsza część komunikatu.

Uwaga: Jako załączniki przywoływane można wysyłać tylko części komunikatu najwyższego poziomu (czyli elementy zdefiniowane w elemencie portType pliku WSDL jako części komunikatu wejściowego lub wyjściowego), które mają typ binarny (base64Binary lub hexBinary).

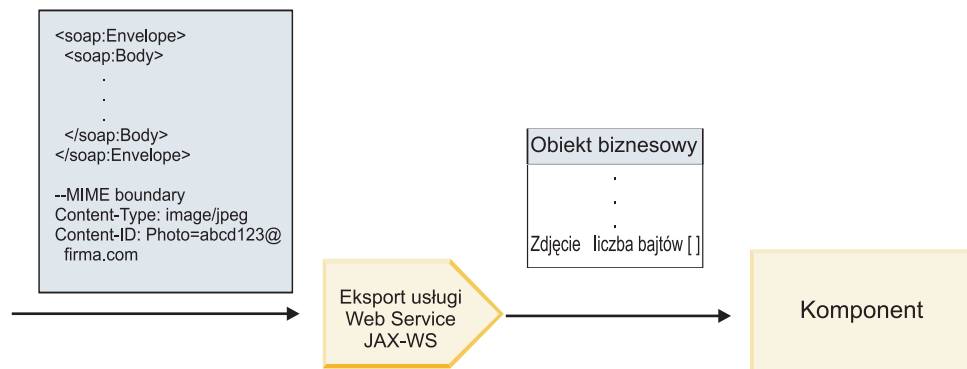
Szczegółowe informacje zawiera temat “Praca z załącznikami” w Centrum informacyjnym produktu Integration Designer.

W przypadku komunikatów zgodnych ze standardem WS-I wygenerowany w komunikacie SOAP identyfikator content-ID jest konkatencją następujących elementów:

- wartości atrybutu **name** elementu **wsdl:part** przywoływanego przez element **mime:content**,
- znaku **=**,
- globalnie unikalnej wartości, na przykład identyfikatora UUID,
- znaku **@**,
- poprawnej nazwy domeny.

Przetwarzanie danych przychodzących załączników przywoływanych

Jeśli klient przekaże komunikat SOAP z załącznikiem do komponentu SCA (Service Component Architecture), powiązanie eksportu usługi Web Service JAX-WS najpierw usuwa załącznik. Następnie analizuje część SOAP komunikatu i tworzy obiekt biznesowy. Na koniec, powiązanie umieszcza binarny załącznik w obiekcie biznesowym.



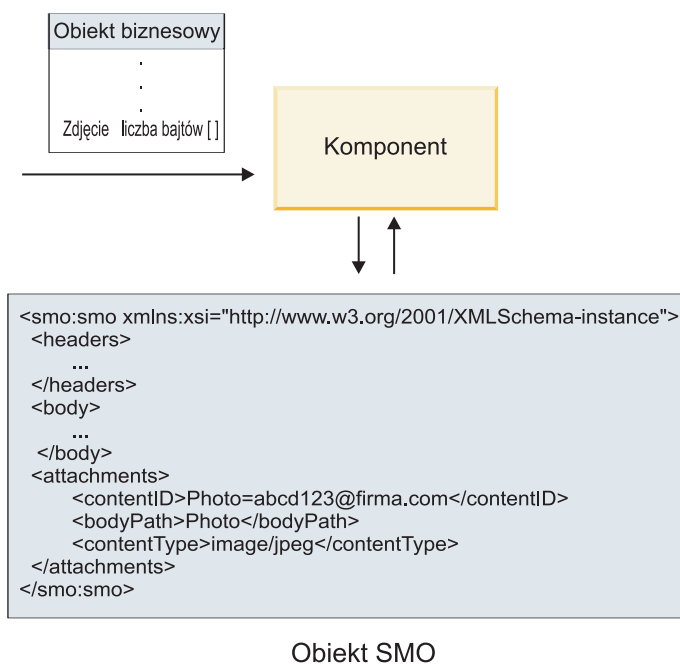
Rysunek 19. Sposób przetwarzania komunikatu SOAP zgodnego ze standardem WS-I i zawierającego załącznik przywoływany przez powiązanie eksportu usługi Web Service JAX-WS

Uzyskiwanie dostępu do metadanych załącznika w komponencie przepływu mediacji

Jak przedstawiono na ilustracji Rys. 19, kiedy komponenty uzyskują dostęp do załączników przywoływanych, dane załącznika są wyświetlane jako tablica bajtów.

Dla każdego załącznika przywoływanego komunikatu SOAP istnieje odpowiadający mu element **attachments** w obiekcie SMO. Element **attachments** zawiera typ treści załącznika oraz ścieżkę do elementu treści komunikatu, w

którym przechowywany jest załącznik.



Rysunek 20. Sposób wyświetlania załączników przywoływanych w obiekcie SMO

Ważne: Ścieżka do elementu treści komunikatu nie jest automatycznie aktualizowana, jeśli komunikat zostanie poddany transformowaniu, a załącznik zostanie przeniesiony. Istnieje możliwość użycia przepływu mediacji do zaktualizowania elementu **attachments** przy użyciu nowej ścieżki (na przykład w ramach transformowania lub za pomocą oddzielnej procedury ustawiającej element komunikatu).

Sposób konstruowania wychodzących komunikatów SOAP

Do skonfigurowania powiązania importu usługi Web Service JAX-WS w celu wywołania zewnętrznej usługi Web Service używany jest produkt Integration Designer. Powiązanie importu jest skonfigurowane przy użyciu dokumentu WSDL, który opisuje usługę Web Service do wywołania oraz definiuje części komunikatu, które mają być przekazywane jako załączniki. Na stronie Określanie zgodności ze standardem WS-I AP 1.0 produktu Integration Designer można również określić jedną z następujących opcji:

- **Użyj komunikatu SOAP zgodnego ze standardem WS-I AP 1.0**

Jeśli zostanie wybrana ta opcja, użytkownik określa również która część komunikatu powinna być powiązana z treścią komunikatu SOAP. Wszystkie pozostałe części są powiązane z załącznikami lub nagłówkami. Komunikaty wysyłane przez powiązanie nie obejmują elementów w treści komunikatu SOAP, które odwołują się do załączników. Relacja jest wyrażana za pośrednictwem identyfikatora treści załącznika zawierającego nazwę części komunikatu.

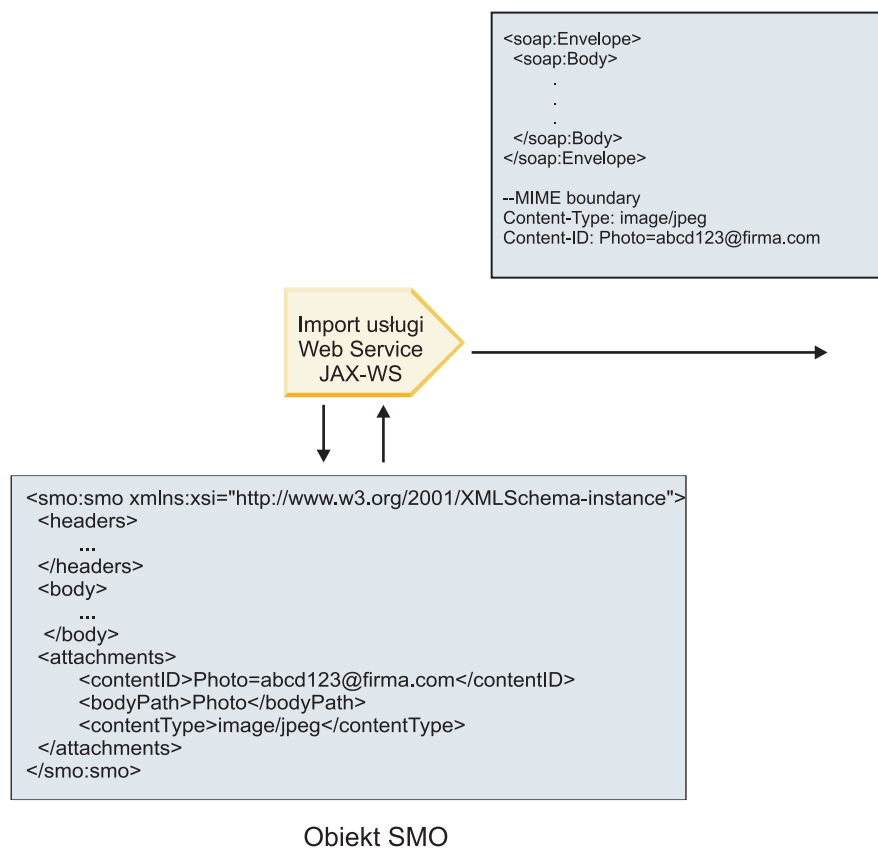
- **Użyj komunikatu SOAP niezgodnego ze standardem WS-I AP 1.0**

Jeśli zostanie wybrana ta opcja (która jest opcją domyślną), z treścią komunikatu SOAP powiązana jest pierwsza część komunikatu. Wszystkie pozostałe części są powiązane z załącznikami lub nagłówkami. Komunikaty wysyłane przez powiązanie zawierają w treści komunikatu SOAP co najmniej jeden element odwołujący się do załączników przy użyciu atrybutu **href**.

Uwaga: Część reprezentująca załącznik, zdefiniowana w dokumencie WSDL, musi być typem prostym (base64Binary lub hexBinary). Jeśli część jest zdefiniowana przy użyciu typu złożonego (complexType), ta część nie może być powiązana jako załącznik.

Przetwarzanie danych wychodzących załączników przywoływanych

Powiązanie importu korzysta z informacji w obiekcie SMO do określania sposobu, w jaki binarne części komunikatu najwyższego poziomu są wysyłane w postaci załączników.



Rysunek 21. Sposób dostępu do załącznika przywoływanego w obiekcie SMO w celu utworzenia komunikatu SOAP

Obiekt SMO zawiera element **attachments** tylko jeśli komponent przepływu mediacji jest połączony bezpośrednio z importem lub eksportem. Element ten nie jest przekazywany przez inne typy komponentów. Jeśli potrzebne są wartości znajdujące się w module zawierającym inne typy komponentów, należy użyć komponentu przepływu mediacji w celu skopiowania tych wartości do miejsca, w którym będzie możliwe uzyskiwanie do nich dostępu z poziomu modułu. Należy również użyć innego komponentu przepływu mediacji do ustawienia poprawnych wartości przed zrealizowaniem wywołania wychodzącego za pośrednictwem importu usługi Web Service.

Powiązanie określa sposób wysyłania komunikatu (lub określa, czy komunikat ma zostać wysłany) na podstawie kombinacji następujących warunków:

- Czy istnieje powiązanie WSDL MIME dla binarnej części komunikatu najwyższego poziomu, a jeśli tak, w jaki sposób zdefiniowany jest typ treści.
- Czy w obiekcie SMO istnieje element **attachments**, którego wartość **bodyPath** odwołuje się do binarnej części najwyższego poziomu.

Sposób tworzenia załączników, jeśli w obiekcie SMO istnieje element attachment

Poniższa tabela przedstawia sposób tworzenia i wysyłania załącznika, jeśli obiekt SMO zawiera element **attachment** z wartością **bodyPath** zgodną z częścią nazwy komunikatu:

Tabela 32. Sposób generowania załącznika

Status powiązania WSDL MIME dla binarnej części komunikatu najwyższego poziomu	Sposób tworzenia i wysyłania komunikatu
<p>Jest obecne, z jednym z następujących elementów:</p> <ul style="list-style-type: none"> • brakiem zdefiniowanego typu treści dla części komunikatu, • wieloma zdefiniowanymi typami treści, • typem treści zdefiniowanym przy użyciu znaku wieloznacznego. 	<p>Część komunikatu jest wysyłana jako załącznik.</p> <p>Identyfikator Content-Id jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest generowany.</p> <p>Parametr Content-Type jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest ustawiany na wartość application/octet-stream.</p>
<p>Jest obecne, z pojedynczą treścią (bez znaku wieloznacznego) dla części komunikatu</p>	<p>Część komunikatu jest wysyłana jako załącznik.</p> <p>Identyfikator Content-Id jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest generowany.</p> <p>Parametr Content-Type jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest ustawiany na typ zdefiniowany w elemencie treści WSDL MIME.</p>
<p>Brak</p>	<p>Część komunikatu jest wysyłana jako załącznik.</p> <p>Identyfikator Content-Id jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest generowany.</p> <p>Parametr Content-Type jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest ustawiany na wartość application/octet-stream.</p> <p>Uwaga: Wysyłanie części komunikatów w postaci załączników, jeśli nie są one tak zdefiniowane w dokumencie WSDL, może spowodować utracenie zgodności ze standardem WS-I Attachments Profile 1.0. Należy więc tego unikać, jeśli to możliwe.</p>

Sposób tworzenia załączników, jeśli w obiekcie SMO nie istnieje element attachment

Poniższa tabela przedstawia sposób tworzenia i wysyłania załącznika, jeśli obiekt SMO nie zawiera elementu **attachment** z wartością **bodyPath** zgodną z częścią nazwy komunikatu:

Tabela 33. Sposób generowania załącznika

Status powiązania WSDL MIME dla binarnej części komunikatu najwyższego poziomu	Sposób tworzenia i wysyłania komunikatu
<p>Jest obecne, z jednym z następujących elementów:</p> <ul style="list-style-type: none"> • brakiem zdefiniowanego typu treści dla części komunikatu, • wieloma zdefiniowanymi typami treści, • typem treści zdefiniowanym przy użyciu znaku wieloznacznego. 	<p>Część komunikatu jest wysyłana jako załącznik.</p> <p>Identyfikator Content-Id jest generowany.</p> <p>Parametr Content-Type jest ustawiany na wartość application/octet-stream.</p>
<p>Jest obecne, z pojedynczą treścią (bez znaku wieloznacznego) dla części komunikatu</p>	<p>Część komunikatu jest wysyłana jako załącznik.</p> <p>Identyfikator Content-Id jest generowany.</p> <p>Parametr Content-Type jest ustawiany na typ zdefiniowany w elemencie treści WSDL MIME.</p>

Tabela 33. Sposób generowania załącznika (kontynuacja)

Status powiązania WSDL MIME dla binarnej części komunikatu najwyższego poziomu	Sposób tworzenia i wysyłania komunikatu
Brak	Część komunikatu nie jest wysyłana jako załącznik.

Ważne: Operacja podstawowa mediacji Odzworowanie transformuje komunikaty przy użyciu transformacji XSLT 1.0. Zostało to opisane w sekcji “Reprezentacja XML obiektu SMO”. Transformacja opiera się na serializacji XML obiektu SMO. Operacja podstawowa mediacji Odzworowanie umożliwia określenie elementu głównego serializacji, a element główny dokumentu XML odzwierciedla ten element główny serializacji.

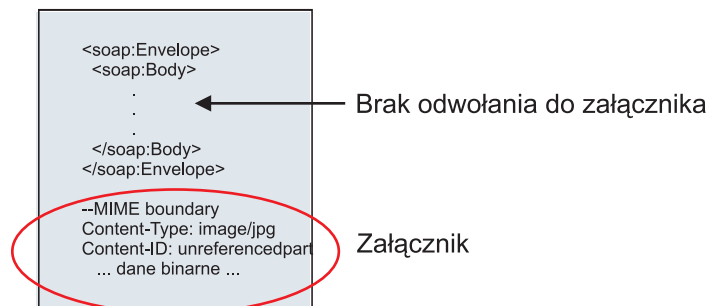
Podczas wysyłania komunikatów SOAP z załącznikami wybrany element główny określa sposób propagowania załączników.

- Jeśli jako element główny odzworowania XML zostanie użyty element “/body”, wszystkie załączniki są domyślnie propagowane do całego odzworowania.
- Jeśli jako element główny odzworowania zostanie użyty element “/”, można kontrolować propagowanie załączników.

Załączniki nieprzywoływane: **Advanced**

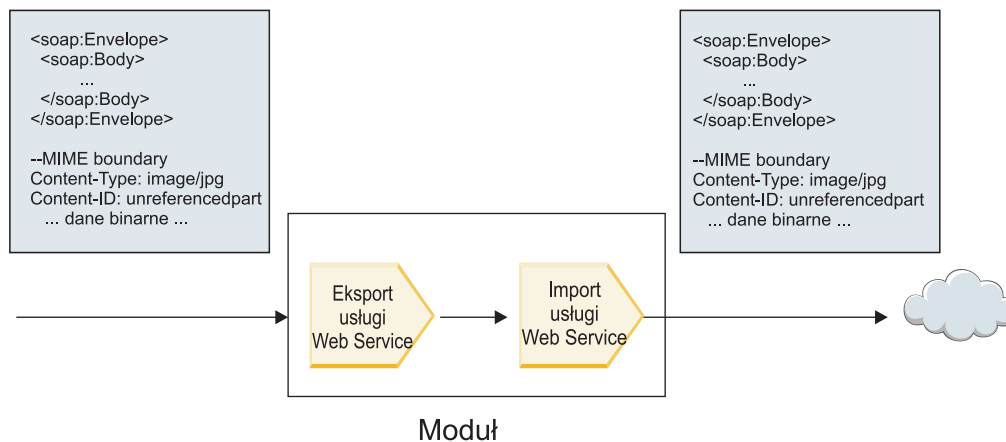
Istnieje możliwość wysyłania i odbierania załączników *nieprzywoływanych*, które nie są deklarowane w interfejsie usługi.

W komunikacie SOAP z wieloma częściami MIME pierwszą częścią komunikatu jest treść SOAP, a załączniki znajdują się w kolejnych częściach. W treści SOAP nie występuje żadne odwołanie do załącznika.



Rysunek 22. Komunikat SOAP z załącznikiem nieprzywoływanym

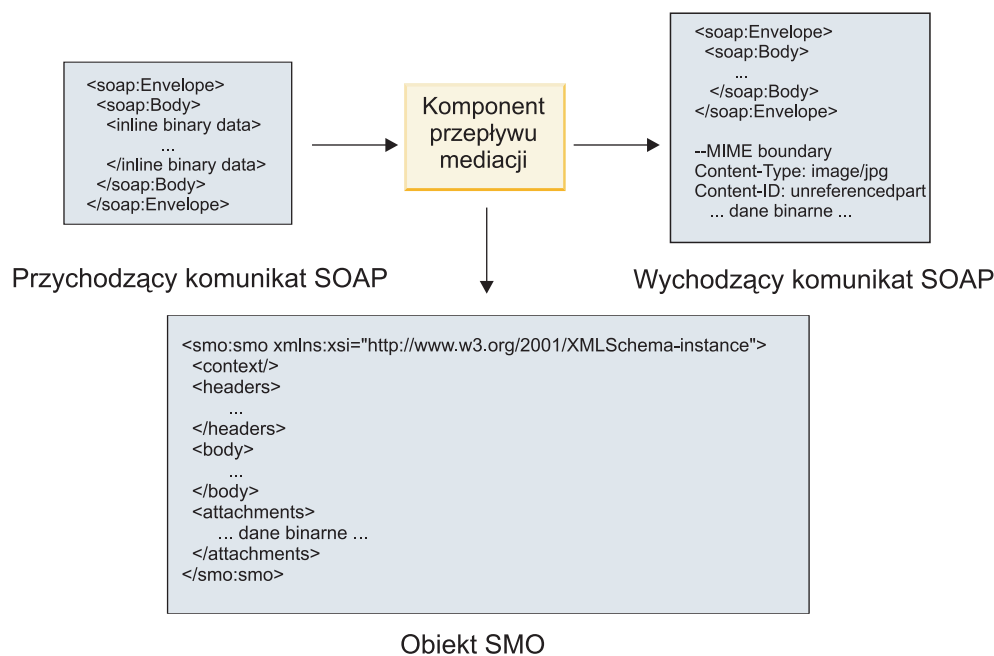
Komunikat SOAP z załącznikiem nieprzywoływanym można wysłać za pośrednictwem eksportu usługi Web Service do importu usługi Web Service. Załącznik jest zawarty w komunikacie wyjściowym wysyłanym do docelowej usługi Web Service.



Rysunek 23. Załącznik przechodzący przez moduł SCA

Na ilustracji Rys. 23 przedstawiono komunikat SOAP z załącznikiem, który jest przekazywany bez modyfikowania.

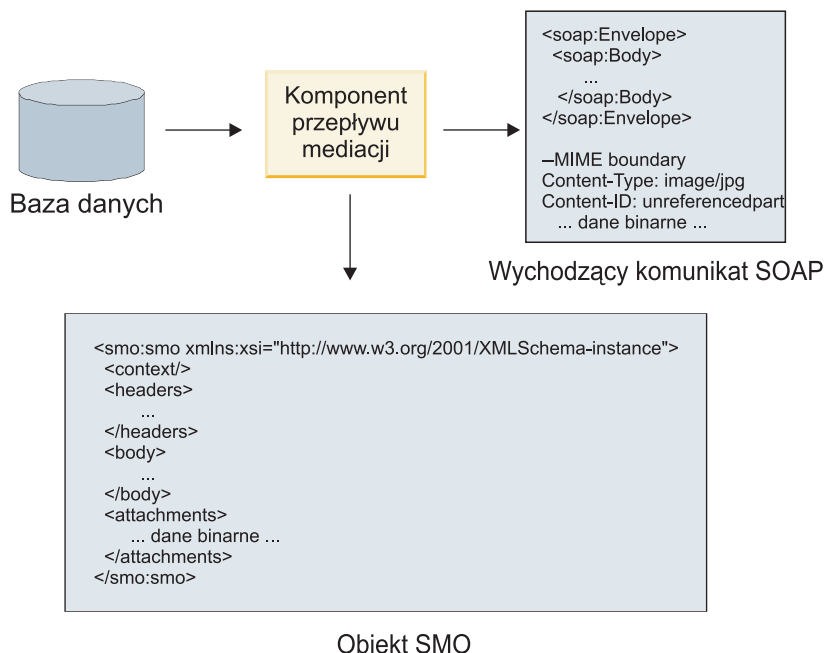
Istnieje również możliwość zmodyfikowania komunikatu SOAP przy użyciu komponentu przepływu mediacji. Można na przykład użyć komponentu przepływu mediacji do wyodrębnienia danych z komunikatu SOAP (w tym przypadku - danych binarnych zawartych w treści komunikatu) i utworzenia komunikatu SOAP z załącznikami. Dane są przetwarzane w ramach elementu załączników obiektu SMO.



Rysunek 24. Komunikat przetworzony przez komponent przepływu mediacji

Komponent przepływu mediacji może również transformować komunikat przychodzący, wyodrębniając i kodując załącznik, a następnie przesyłając komunikat bez załączników.

Zamiast wyodrębniania danych z przychodzącego komunikatu SOAP w celu utworzenia komunikatu SOAP z załącznikami, można uzyskać dane załącznika ze źródła zewnętrznego, na przykład bazy danych.



Rysunek 25. Załącznik uzyskany z bazy danych i dodany do komunikatu SOAP

Komponent przepływu mediacji może również wyodrębnić załącznik z przychodzącego komunikatu SOAP i przetworzyć ten komunikat (na przykład zapisując załącznik w bazie danych).

Załączniki nieprzywoływane mogą być propagowane tylko przez komponenty przepływu mediacji. Jeśli załącznik wymaga dostępu z poziomu innego typu komponentu lub propagowania do innego typu komponentu, należy użyć komponentu przepływu mediacji w celu przeniesienia załącznika do miejsca, które jest dostępne dla tego komponentu.

Ważne: Operacja podstawowa mediacji Odzworowanie transformuje komunikaty przy użyciu transformacji XSLT 1.0. Zostało to opisane w sekcji “Reprezentacja XML obiektu SMO”. Transformacja opiera się na serializacji XML obiektu SMO. Operacja podstawowa mediacji Odzworowanie umożliwia określenie elementu głównego serializacji, a element główny dokumentu XML odzwierciedla ten element główny serializacji.

Podczas wysyłania komunikatów SOAP z załącznikami wybrany element główny określa sposób propagowania załączników.

- Jeśli jako element główny odwzorowania XML zostanie użyty element “/body”, wszystkie załączniki są domyślnie propagowane do całego odwzorowania.
- Jeśli jako element główny odwzorowania zostanie użyty element “/”, można kontrolować propagowanie załączników.

Używanie powiązania stylu dokumentu WSDL z komunikatami wieloczęściowymi: **Advanced**

Organizacja WS-I (Web Services Interoperability Organization) zdefiniowała zbiór reguł sposobu opisywania usług Web Services za pomocą dokumentu WSDL oraz sposobu formułowania odpowiednich komunikatów SOAP w celu zagwarantowania współdziałania.

Te reguły są określone w standardzie WS-I *Basic Profile Version 1.1* (profil podstawowy, wersja 1.1 - <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>). W szczególności dokument WS-I Basic Profile 1.1 R2712 określa, że: „Powiązanie w stylu document-literal MUSI być serializowane jako element ENVELOPE z elementem soap:Body, którego element potomny jest instancją deklaracji elementu globalnego przywoływanej przez odpowiednią część wsdl:message”.

Oznacza to, że podczas używania powiązania SOAP w stylu document w operacji z komunikatami (wejściowymi, wyjściowymi lub błędu) zdefiniowanymi z wieloma częściami, tylko jedna z tych części powinna być powiązana z treścią komunikatu SOAP, aby zachować zgodność z profilem WS-I Basic Profile 1.1.

Dodatkowo dokument WS-I Attachments Profile 1.0 R2941 określa, że: „Element wsdl:binding w elemencie DESCRIPTION POWINIEN wiązać każdy element wsdl:part elementu wsdl:message w elemencie wsdl:portType, do którego się odwołuje, z jednym z elementów: soapbind:body, soapbind:header, soapbind:fault, soapbind:headerfault lub mime:content”.

Oznacza to, że podczas używania powiązania SOAP w stylu document w operacji z komunikatami (wejściowymi, wyjściowymi lub błędu) zdefiniowanymi z wieloma częściami, wszystkie części inne niż ta, która została wybrana do powiązania z treścią komunikatu SOAP, powinny być powiązane jako załączniki lub nagłówki.

Podczas generowania opisów WSDL na potrzeby eksportów z powiązaniem usług Web Services (JAX-WS i JAX-RPC) jest w tym przypadku stosowane następujące podejście:

- Użytkownik może wybrać, która część komunikatu jest powiązana z treścią komunikatu SOAP, jeśli istnieje więcej niż jeden element typu innego niż binarny. Jeśli istnieje tylko jeden element typu innego niż binarny, ten element jest automatycznie powiązany z treścią komunikatu SOAP.
- W przypadku powiązania JAX-WS wszystkie inne części komunikatu typu hexBinary lub base64Binary są powiązane jako załączniki przywoływane. Patrz temat “Załączniki przywoływane: części komunikatu najwyższego poziomu” na stronie 87.
- Wszystkie inne części komunikatu są powiązane jako nagłówki SOAP.

Powiązania importu JAX-RPC i JAX-WS uwzględniają powiązanie SOAP w istniejącym dokumencie WSDL zawierającym komunikaty w stylu multipart document, nawet jeśli nie wiąże on wielu części z treścią komunikatu SOAP. Nie ma jednak możliwości wygenerowania klientów usług Web Services dla takich dokumentów WSDL w produkcie Rational Application Developer.

Uwaga: Powiązanie JAX-RPC nie obsługuje załączników.

Zalecany wzorec podczas używania komunikatów wieloczęściowych z operacją, która ma powiązanie SOAP w stylu document jest więc następujące podejście:

1. Używanie stylu document/literal wrapped. W tym przypadku komunikaty zawsze mają jedną część, ale załączniki muszą być nieprzywoływane (jak opisano w temacie “Załączniki nieprzywoływane” na stronie 92) lub muszą mieć typ swaRef (jak opisano w temacie “Załączniki przywoływane: elementy typu swaRef” na stronie 83).
2. Używanie stylu RPC/literal. W tym przypadku nie ma ograniczeń powiązania WSDL w zakresie liczby części powiązanych z treścią komunikatu SOAP. Wynikowe komunikaty SOAP zawsze mają pojedynczy element potomny reprezentujący wywoływaną operację, a części komunikatu są elementami potomnymi tego elementu.
3. W przypadku powiązania JAX-WS - stosowanie maksymalnie jednej części komunikatu, która nie ma typu hexBinary ani base64Binary, chyba że dopuszczalne jest powiązanie pozostałych części innych niż binarne z nagłówkami SOAP.
4. Wszystkie inne przypadki podlegają opisanemu sposobowi działania.

Uwaga: Istnieją dodatkowe ograniczenia używania komunikatów SOAP niezgodnych z profilem WS-I *Basic Profile Version 1.1*.

- Pierwsza część komunikatu powinna być inna niż binarna.
- Podczas odbierania komunikatów SOAP w stylu multipart document z załącznikami przywoływanymi powiązanie JAX-WS oczekuje, że każdy załącznik przywoływany będzie reprezentowany przez element potomny treści komunikatu SOAP z wartością atrybutu href identyfikującą załącznik na podstawie jego identyfikatora treści. Powiązanie JAX-WS wysyła załączniki przywoływane dla takich komunikatów, wykorzystując ten sam sposób. To działanie nie jest zgodne z podstawowym profilem WS-I.

Aby zagwarantować, że komunikaty są zgodne z profilem podstawowym, należy postępować zgodnie z punktem 1 na stronie 95 lub punktem 2 na stronie 95 na wcześniej liście lub unikać używania załączników przywoływanych dla takich komunikatów i zamiast nich użyć załączników nieprzywoływanych lub załączników typu swaRef.

Powiązania HTTP:

Powiązanie HTTP zostało zaprojektowane, aby udostępniać architekturze SCA (Service Component Architecture) połączenia z protokołem HTTP. Dzięki niemu istniejące lub nowo utworzone aplikacje HTTP mogą uczestniczyć w środowiskach architektury zorientowanej na usługi (SOA).

Protokół HTTP (Hypertext Transfer Protocol) jest powszechnie używanym protokołem służącym do przesyłania informacji w sieci WWW. Jeśli użytkownik pracuje z zewnętrzną aplikacją używającą protokołu HTTP, niezbędne jest zastosowanie powiązania HTTP. Powiązanie HTTP transformuje dane przekazywane jako komunikat w formacie rodzimym do obiektu biznesowego w aplikacji SCA. Powiązanie HTTP może również transformować dane przekazywane jako obiekt biznesowy na format rodzimy oczekiwany przez aplikację zewnętrzną.

Uwaga: Aby prowadzić interakcję z klientami i usługami korzystającymi z protokołu SOAP/HTTP usług Web Services, należy rozważyć użycie jednego z powiązań usługi Web Service. Powiązania te udostępniają dodatkową funkcjonalność, umożliwiającą obsługiwanie standardowych elementów jakości usług w usługach Web Services.

Niektóre typowe scenariusze używania powiązania HTTP zostały wymienione na poniższej liście:

- Usługi udostępniane przez architekturę SCA mogą wywoływać aplikacje HTTP przy użyciu importu HTTP.
- Usługi udostępniane przez architekturę SCA mogą ujawniać się jako aplikacje z funkcjami HTTP, aby mogły być używane przez klienty HTTP za pośrednictwem eksportu HTTP.
- IBM Business Process Manager i Process Server mogą komunikować się ze sobą za pośrednictwem infrastruktury HTTP, a więc użytkownicy mogą zarządzać komunikacją przy zastosowaniu standardów korporacyjnych.
- IBM Business Process Manager i Process Server mogą działać jako mediatory komunikacji HTTP, transformując i kierując komunikaty, co usprawnia integrację aplikacji przy użyciu sieci HTTP.
- IBM Business Process Manager i Process Server mogą być używane do utworzenia mostu między protokołem HTTP i innymi protokołami takimi jak protokół SOAP/HTTP usług Web Services, adaptery oparte na zasobach JCA (Java Connector Architecture), protokół JMS itd.

Szczegółowe informacje na temat tworzenia powiązań importu i eksportu HTTP zawiera Centrum informacyjne produktu Integration Designer. Należy zapoznać się z tematami **Tworzenie aplikacji integracji > Uzyskiwanie dostępu do usług zewnętrznych przy użyciu protokołu HTTP**.

Przegląd powiązań HTTP:

Powiązanie HTTP udostępnia połączenia z aplikacjami udostępnianymi za pośrednictwem protokołu HTTP. Mediuje ono komunikację między aplikacjami HTTP i umożliwia wywoływanie istniejących aplikacji opartych na protokole HTTP z modułu.

Powiązania importu HTTP

Powiązanie importu HTTP udostępnia połączenia wychodzące z aplikacji SCA (Service Component Architecture) do serwera HTTP lub aplikacji HTTP.

Import wywołuje adres URL punktu końcowego protokołu HTTP. Adres URL można określić na jeden z trzech sposobów:

- Adres URL może być ustawiany dynamicznie w nagłówkach HTTP za pośrednictwem dynamicznie nadpisywanego adresu URL.
- Adres URL może być ustawiany dynamicznie w elemencie docelowego adresu obiektu SMO.
- Adres URL może być określany jako właściwość konfiguracji w imporcie.

To wywołanie jest zawsze synchroniczne.

Mimo że wywołania HTTP są zawsze wywołaniami typu żądanie-odpowiedź, import HTTP obsługuje zarówno operacje jednokierunkowe, jak i dwukierunkowe oraz ignoruje odpowiedzi w przypadku operacji jednokierunkowych.

Powiązania eksportu HTTP

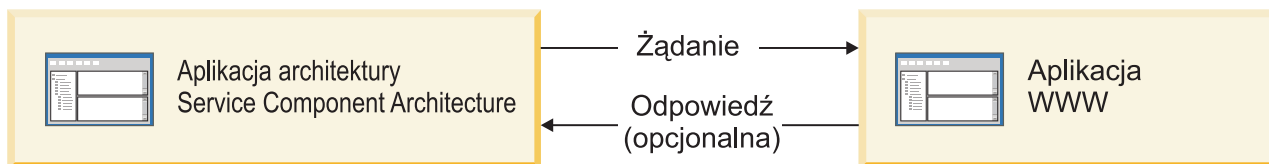
Powiązanie eksportu HTTP udostępnia połączenia przychodzące z aplikacji HTTP do aplikacji SCA.

Adres URL jest definiowany w eksporcie HTTP. Aplikacje HTTP, które wymagają wysyłania komunikatów żądania do eksportu, używają tego adresu URL do wywołania eksportu.

Eksport HTTP obsługuje również komendy ping.

Powiązania HTTP w czasie wykonywania

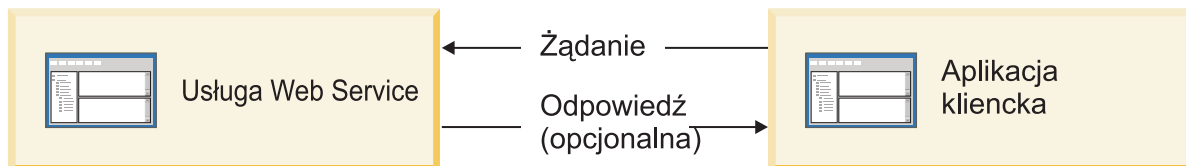
W czasie wykonywania import z powiązaniem HTTP wysyła żądanie z aplikacji SCA do zewnętrznej usługi Web Service, zawierające dane w treści komunikatu lub niezawierające tych danych. Żądanie jest wykonywane z aplikacji SCA do zewnętrznej usługi Web Service. Zostało to przedstawione na następującej ilustracji: Rys. 26.



Rysunek 26. Przepływ żądania z aplikacji SCA do aplikacji WWW

Opcjonalnie import z powiązaniem HTTP może odbierać dane z aplikacji WWW (w odpowiedzi na żądanie).

W przypadku eksportu żądanie jest wykonywane przez aplikację kliencką do usługi Web Service. Zostało to przedstawione na następującej ilustracji: Rys. 27.



Rysunek 27. Przepływ żądania z aplikacji klienckiej do usługi Web Service

Usługa Web Service jest aplikacją WWW działającą na serwerze. Eksport jest implementowany w tej aplikacji WWW w postaci serwletu, a więc klient wysyła żądanie do adresu URL. Serwlet przekazuje żądanie do aplikacji SCA w środowisku wykonawczym.

Opcjonalnie eksport może wysyłać dane do aplikacji klienckiej (w odpowiedzi na żądanie).

Nagłówki HTTP:

Powiązania importu i eksportu HTTP pozwalają, aby konfiguracja nagłówków HTTP oraz ich wartości były używane na potrzeby komunikatów wychodzących. Import HTTP używa tych nagłówków na potrzeby żądań, a eksport HTTP używa ich na potrzeby odpowiedzi.

Statycznie skonfigurowane nagłówki i informacje kontrolne są nadrzędne w stosunku do wartości ustawianych dynamicznie w czasie wykonywania. Jednak dynamicznie nadpisywane wartości kontrolne adresu URL, wersji i metody przesłaniają wartości statyczne, które są w przeciwnym razie uznawane za domyślne.

Powiązanie obsługuje dynamiczny rodzaj adresu URL importu HTTP, określając wartość docelowego adresu URL protokołu HTTP, wersję i metodę w czasie wykonywania. Te wartości są określane poprzez wyodrębnianie wartości odwołania punktu końcowego, dynamicznie nadpisywanego adresu URL, wersji oraz metody.

- W przypadku odwołania punktu końcowego należy użyć funkcji API `com.ibm.websphere.sca.addressing.EndpointReference` lub ustawić pole `/headers/SMOHeader/Target/address` w nagłówku obiektu SMO.
- W przypadku dynamicznie nadpisywanego adresu URL, wersji i metody, należy użyć sekcji parametrów sterujących protokołu HTTP w komunikacie SCA (Service Component Architecture). Należy zauważyć, że dynamicznie nadpisywany adres URL ma pierwszeństwo w kolejności wykonywania operacji przed docelowym odwołaniem punktu końcowego. Z kolei odwołanie punktu końcowego ma zastosowanie w wielu powiązaniach, a więc jest uznawane za preferowane podejście i należy go używać tam, gdzie to możliwe.

Informacje kontrolne i informacje w nagłówku dla komunikatów wychodzących w powiązaniach eksportu i importu HTTP są przetwarzane w następującej kolejności:

1. Informacje w nagłówku i informacje kontrolne z wyjątkiem dynamicznie nadpisywanego adresu URL protokołu HTTP, wersji i metody z komunikatu SCA (najniższy priorytet).
2. Zmiany dokonane w Konsoli administracyjnej na poziomie eksportu/importu.
3. Zmiany dokonane w Konsoli administracyjnej na poziomie metody eksportu lub importu.
4. Adres docelowy określony za pośrednictwem odwołania punktu końcowego lub nagłówka obiektu SMO.
5. Dynamicznie nadpisywany adres URL, wersja i metoda z komunikatu SCA.
6. Informacje w nagłówku i informacje kontrolne z procedury obsługi danych lub powiązania danych (najwyższy priorytet).

Eksport i import HTTP wypełniają nagłówki wychodzące i parametry sterujące danymi z komunikatu przychodzącego (HTTPExportRequest i HTTPImportResponse) tylko, jeśli propagacja nagłówków protokołów jest ustawiona na wartość **True**. Eksport i import HTTP odczytują i przetwarzają nagłówki wychodzące i parametry sterujące (HTTPExportResponse i HTTPImportRequest) tylko, jeśli propagacja nagłówków protokołów jest ustawiona na wartość **True**.

Uwaga: Zmiany wprowadzone przez procedury obsługi danych lub powiązania danych do nagłówków lub parametrów sterujących w odpowiedzi importu lub żądaniu eksportu nie będą miały wpływu na instrukcje przetwarzania komunikatu wewnątrz powiązania importu lub eksportu i powinny być używane tylko w celu propagowania zmodyfikowanych wartości do zagnieżdżonych komponentów SCA.

W ramach usługi kontekstu jest wykonywane propagowanie kontekstu (w tym nagłówków protokołów, takich jak nagłówek HTTP, i kontekstu użytkownika, na przykład identyfikatora konta) w ścieżce wywołań SCA. Podczas programowania w produkcie IBM Integration Designer użytkownik może kontrolować propagowanie kontekstu za pośrednictwem właściwości importu i eksportu. Więcej szczegółów można znaleźć w informacjach na temat powiązań importów i eksportów w Centrum informacyjnym produktu IBM Integration Designer.

Dostarczane struktury nagłówka HTTP i obsługa

Tabela Tabela 34 przedstawia parametry żądania/odpowiedzi dla żądań i odpowiedzi importu HTTP i eksportu HTTP.

Tabela 34. Dostarczane informacje nagłówka HTTP

Nazwa elementu sterującego	Żądanie importu HTTP	Odpowiedź importu HTTP	Żądanie eksportu HTTP	Odpowiedź eksportu HTTP
URL (adres URL)	Zignorowano	Nieustawiony	Odczytywany z komunikatu żądania. Uwaga: Łańcuch zapytania również stanowi część parametru sterującego adresu URL.	Zignorowano

Tabela 34. Dostarczane informacje nagłówka HTTP (kontynuacja)

Nazwa elementu sterującego	Żądanie importu HTTP	Odpowiedź importu HTTP	Żądanie eksportu HTTP	Odpowiedź eksportu HTTP
Version (wersja; możliwe wartości: 1.0, 1.1; wartość domyślna to 1.1)	Zignorowano	Nieustawiony	Odczytywany z komunikatu żądania	Zignorowano
Method (metoda)	Zignorowano	Nieustawiony	Odczytywany z komunikatu żądania	Zignorowano
Dynamic Override URL (dynamicznie nadpisywany adres URL)	Jeśli jest ustawiony w procedurze obsługi danych lub powiązaniu danych, nadpisuje adres URL importu HTTP. Zapisywany w komunikacie w wierszu żądania. Uwaga: Łańcuch zapytania również stanowi część parametru sterującego adresu URL.	Nieustawiony	Nieustawiony	Zignorowano
Dynamic Override Version (dynamicznie nadpisywana wersja)	Jeśli jest ustawiony, nadpisuje wersję importu HTTP. Zapisywany w komunikacie w wierszu żądania.	Nieustawiony	Nieustawiony	Zignorowano
Dynamic Override Method (dynamicznie nadpisywana metoda)	Jeśli jest ustawiony, nadpisuje metodę importu HTTP. Zapisywany w komunikacie w wierszu żądania.	Nieustawiony	Nieustawiony	Zignorowano
Media Type (typ nośnika; ten parametr sterujący przenosi część wartości nagłówka HTTP Content-Type)	Jeśli jest obecny, jest zapisywany w komunikacie jako część nagłówka Content-Type. Uwaga: Wartość tego elementu sterującego powinna być udostępniana przez procedurę obsługi danych lub powiązanie danych.	Odczytywany z komunikatu odpowiedzi, nagłówka Content-Type	Odczytywany z komunikatu żądania, nagłówka Content-Type	Jeśli jest obecny, jest zapisywany w komunikacie jako część nagłówka Content-Type. Uwaga: Wartość tego elementu sterującego powinna być udostępniana przez procedurę obsługi danych lub powiązanie danych.
Character Set (zestaw znaków, domyślnie: UTF-8)	Jeśli jest obecny, jest zapisywany w komunikacie jako część nagłówka Content-Type. Uwaga: Wartość tego elementu sterującego powinna być udostępniana przez powiązanie danych.	Odczytywany z komunikatu odpowiedzi, nagłówka Content-Type	Odczytywany z komunikatu żądania, nagłówka Content-Type	Obsługiwany: zapisywany w komunikacie jako część nagłówka Content-Type. Uwaga: Wartość tego elementu sterującego powinna być udostępniana przez powiązanie danych.

Tabela 34. Dostarczane informacje nagłówka HTTP (kontynuacja)

Nazwa elementu sterującego	Żądanie importu HTTP	Odpowiedź importu HTTP	Żądanie eksportu HTTP	Odpowiedź eksportu HTTP
Transfer Encoding (kodowanie przesyłanych danych, możliwe wartości: chunked, identity; wartość domyślna: identity)	Jeśli jest obecny, jest zapisywany w komunikacie jako nagłówek i steruje kodowaniem transformowania komunikatu.	Odczytywany z komunikatu odpowiedzi	Odczytywany z komunikatu żądania	Jeśli jest obecny, jest zapisywany w komunikacie jako nagłówek i steruje kodowaniem transformowania komunikatu.
Content Encoding (kodowanie treści, możliwe wartości: gzip, x-gzip, deflate, identity; wartość domyślna: identity)	Jeśli jest obecny, jest zapisywane w komunikacie jako nagłówek i steruje kodowaniem ładunku.	Odczytywany z komunikatu odpowiedzi	Odczytywany z komunikatu żądania	Jeśli jest obecny, jest zapisywane w komunikacie jako nagłówek i steruje kodowaniem ładunku.
Content-Length (długość treści)	Zignorowano	Odczytywany z komunikatu odpowiedzi	Odczytywany z komunikatu żądania	Zignorowano
StatusCode (kod statusu, wartość domyślna: 200)	Nieobsługiwane	Odczytywany z komunikatu odpowiedzi	Nieobsługiwane	Jeśli jest obecny, jest zapisywany w komunikacie w wierszu odpowiedzi.
ReasonPhrase (frazą przyczyny, wartość domyślna: OK)	Nieobsługiwane	Odczytywany z komunikatu odpowiedzi	Nieobsługiwane	Wartość kontrolna jest ignorowana. Wartość wiersza odpowiedzi komunikatu jest generowana na podstawie kodu statusu.
Authentication (uwierzytelnianie, zawiera wiele właściwości)	Jeśli jest obecny, jest używany do konstruowania nagłówka podstawowego uwierzytelnienia. Uwaga: Wartość tego nagłówka będzie kodowana tylko w protokole HTTP. W architekturze SCA będzie on dekodowany i przekazywany w formie jawnego tekstu.	Nie dotyczy	Odczytywany z nagłówka podstawowego uwierzytelnienia komunikatu żądania. Obecność tego nagłówka nie oznacza, że użytkownik został uwierzytelniony. Uwierzytelnianie powinno być kontrolowane w konfiguracji serwletu. Uwaga: Wartość tego nagłówka będzie kodowana tylko w protokole HTTP. W architekturze SCA będzie on dekodowany i przekazywany w formie jawnego tekstu.	Nie dotyczy
Proxy (zawiera wiele właściwości: Host, Port, Authentication - uwierzytelnianie)	Jeśli jest obecny, jest używany do nawiązywania połączenia za pośrednictwem serwera proxy.	Nie dotyczy	Nie dotyczy	Nie dotyczy

Tabela 34. Dostarczane informacje nagłówka HTTP (kontynuacja)

Nazwa elementu sterującego	Żądanie importu HTTP	Odpowiedź importu HTTP	Żądanie eksportu HTTP	Odpowiedź eksportu HTTP
SSL (zawiera wiele właściwości: Keystore - magazyn kluczy, Keystore Password - hasło magazynu kluczy, Trustore - zaufany magazyn kluczy, Trustore Password - hasło zaufanego magazynu kluczy, ClientAuth - uwierzytelnianie klienta)	Jeśli jest wypełniony, a docelowy adres URL jest adresem protokołu HTTPS, jest używany do nawiązania połączenia za pośrednictwem protokołu SSL.	Nie dotyczy	Nie dotyczy	Nie dotyczy

Powiązania danych HTTP:

Dla każdego odwzorowania danych między komunikatem SCA (Service Component Architecture) i komunikatem protokołu HTTP należy skonfigurować procedurę obsługi danych lub powiązanie danych HTTP. Procedury obsługi danych udostępniają interfejs niezależny od powiązania, który może być ponownie wykorzystywany w powiązaniach transportu i stanowi rekomendowane podejście. Powiązania danych są specyficzne dla konkretnego powiązania transportu. Klasy powiązań danych specyficznych dla protokołu HTTP są dostarczane wraz z produktem. Użytkownik może również tworzyć niestandardowe procedury obsługi danych lub powiązania danych.

Uwaga: Trzy klasy powiązań danych HTTP opisywane w niniejszym temacie (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML i HTTPServiceGatewayDataBinding) są nieaktualne w produkcie IBM Business Process Manager 7.0. Zamiast używać powiązań danych opisanych w tym temacie warto rozważyć zastosowanie następujących procedur obsługi danych:

- SOAPDataHandler zamiast HTTPStreamDataBindingSOAP,
- UTF8XMLDataHandler zamiast HTTPStreamDataBindingXML,
- GatewayTextDataHandler zamiast HTTPServiceGatewayDataBinding.

Do używania z importami i eksportami HTTP udostępniane są powiązania danych: powiązanie danych binarnych, powiązanie danych XML i powiązanie danych SOAP. Powiązanie danych odpowiedzi nie jest wymagane w przypadku operacji jednokierunkowych. Powiązanie danych jest reprezentowane przez nazwę klasy Java, której instancje mogą dokonywać konwersji z protokołu HTTP na obiekt ServiceDataObject i z powrotem. W eksporcie należy użyć selektora funkcji, który w połączeniu z powiązaniem metod może określić używane powiązanie danych i wywołaną operację. Dostarczane są następujące powiązania danych:

- Powiązania danych binarnych, które traktują treść jako nieustrukturyzowane dane binarne. Implementacja schematu XSD powiązania danych binarnych jest następująca:

```
<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>
```

```
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

- Powiązania danych XML, które obsługują treść jako dane XML. Implementacja powiązania danych XML jest podobna do powiązania danych XML JMS i nie ma żadnych ograniczeń schematu interfejsu.
- Powiązania danych SOAP, które obsługują treść jako dane SOAP. Implementacja powiązania danych SOAP nie ma ograniczeń schematu interfejsu.

Implementowanie niestandardowych powiązań danych HTTP

Ta sekcja opisuje, w jaki sposób implementowane są niestandardowe powiązania danych HTTP.

Uwaga: Zalecany podejściem jest implementowanie niestandardowej procedury obsługi danych, ponieważ może być ona ponownie wykorzystywana w powiązaniach transportu.

Podstawowym interfejsem do obsługi niestandardowych komunikatów HTTP jest interfejs `HTTPStreamDataBinding`. Interfejs ten jest zaprojektowany tak, aby umożliwić obsługę dużych ładunków. Jednak aby taka implementacja działała, powiązanie danych musi zwracać informacje kontrolne i nagłówki przed zapisywaniem komunikatu do strumienia.

Niestandardowe powiązanie danych musi implementować metody i kolejność wykonywania zgodnie z poniższą listą.

Aby dostosować powiązanie danych, należy napisać klasę implementującą interfejs `HTTPStreamDataBinding`. Powiązanie danych powinno mieć cztery właściwości prywatne:

- `private DataObject pDataObject`,
- `private HTTPControl pCtrl`,
- `private HTTPHeaders pHeaders`,
- `private yourNativeDataType nativeData`.

Powiązanie HTTP będzie wywoływało niestandardowe powiązanie danych w następującej kolejności:

- Przetwarzanie danych wychodzących (obiekt `DataObject` do formatu rodzimego):
 1. `setDataObject(...)`
 2. `setHeaders(...)`
 3. `setControlParameters(...)`
 4. `setBusinessException(...)`
 5. `convertToNativeData()`
 6. `getControlParameters()`
 7. `getHeaders()`
 8. `write(...)`
- Przetwarzanie danych przychodzących (format rodzimy do obiektu `DataObject`):
 1. `setControlParameters(...)`
 2. `setHeaders(...)`
 3. `convertFromNativeData(...)`
 4. `isBusinessException()`
 5. `getDataObject()`
 6. `getControlParameters()`
 7. `getHeaders()`

Metoda `setDataObject(...)` musi być wywołana w metodzie `convertFromNativeData(...)`, aby ustawić wartość obiektu `dataObject`, który jest przekształcany z danych rodzimych we właściwość prywatną `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;
}
/*
 * Dodawanie nagłówka HTTP IsBusinessException w nagłówkach pHeaders.
 * Dwa kroki:
 * 1.Usunięcie wszystkich nagłówków o nazwie IsBusinessException (bez rozróżniania wielkości liter).
 *   Ma to na celu upewnienie się, że istnieje tylko jeden taki nagłówek.
 * 2.Dodanie nowego nagłówka IsBusinessException.
 */
public void setBusinessException(boolean isBusinessException) {
    //Usunięcie wszystkich nagłówków o nazwie IsBusinessException (bez rozróżniania wielkości liter).
    //Ma to na celu upewnienie się, że istnieje tylko jeden taki nagłówek.
    //Dodanie nowego nagłówka IsBusinessException. Przykład kodu:
    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;
}
/*
 * Uzyskanie nagłówka IsBusinessException z nagłówków pHeaders, zwrócenie jego wartości boolowskiej.
 */
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){
    //Metoda opracowana przez klienta:
    //Odczytywanie danych z interfejsu HTTPInputStream.
    //Przekształcenie danych w obiekt DataObject.
    DataObject dataobject=realConvertWorkFromNativeDataToSD0(arg0);
    setDataObject(dataobject);
}

public void write(HTTPOutputStream output) throws IOException {
    if (nativeData != null)
        output.write(nativeData);
}

```

Powiązania EJB:

Powiązania importu EJB (Enterprise JavaBeans) umożliwiają komponentom SCA (Service Component Architecture) wywoływanie usług udostępnianych przez logikę biznesową Java EE działającą na serwerze Java EE. Powiązania

eksportu EJB umożliwiają ujawnianie komponentów SCA jako komponentów EJB (Enterprise JavaBeans), co pozwala logice biznesowej Java EE wywoływać komponenty SCA, które są w przeciwnym razie dla nich niedostępne.

Powiązania importu EJB:

Powiązania importu EJB umożliwiają modułowi SCA wywoływanie implementacji EJB poprzez określenie sposobu, w jaki moduł konsumenta jest powiązany z zewnętrznym komponentem EJB. Importowanie usług z zewnętrznej implementacji EJB umożliwia użytkownikom podłączenie logiki biznesowej do środowiska produktu IBM Business Process Manager i uczestnictwo w procesie biznesowym.

Do tworzenia powiązań importu EJB służy produkt Integration Designer. Do wygenerowania powiązań można użyć jednej z poniższych procedur:

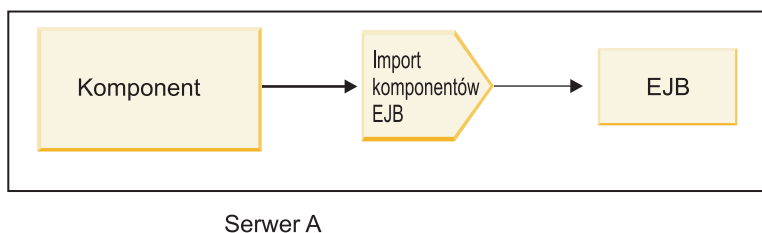
- Tworzenie importu EJB przy użyciu kreatora usług zewnętrznych
Do zbudowania importu EJB na podstawie istniejącej implementacji można użyć kreatora usług zewnętrznych w produkcie Integration Designer. Kreator usług zewnętrznych umożliwia tworzenie usług na podstawie podawanych przez użytkownika kryteriów. Generuje on następnie obiekty biznesowe, interfejsy oraz pliki importu w oparciu o wykryte usługi.
- Tworzenie importu EJB przy użyciu edytora składania
Istnieje możliwość utworzenia importu EJB w diagramie składania przy użyciu edytora składania produktu Integration Designer. Do utworzenia powiązania EJB przy użyciu palety można użyć importu lub klasy Java.

Wygenerowany import ma powiązania danych, które nawiązują połączenie Java-WSDL i nie wymagają komponentu pomocowego języka Java. Komponent można połączyć bezpośrednio z odwołaniem WSDL (Web Services Description Language) prowadzącym do importu EJB komunikującego się z usługą opartą na komponencie EJB przy użyciu interfejsu Java.

Import EJB może realizować interakcję z logiką biznesową Java EE przy użyciu modelu programowania EJB 2.1 lub modelu programowania EJB 3.0.

Wywołanie logiki biznesowej Java EE może być lokalne (tylko dla modelu EJB 3.0) lub zdalne.

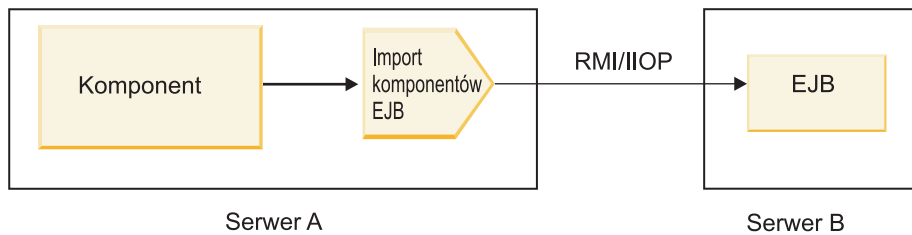
- Wywołanie lokalne jest używane, jeśli istnieje potrzeba wywołania logiki biznesowej Java EE znajdującej się na tym samym serwerze co import.



Rysunek 28. Wywołanie lokalne komponentu EJB (tylko dla modelu EJB 3.0)

- Wywołanie zdalne jest używane, jeśli istnieje potrzeba wywołania logiki biznesowej Java EE, która nie znajduje się na tym samym serwerze co import.

Na poniższej przykładowej ilustracji import EJB używa zdalnego wywołania metody za pośrednictwem protokołu Internet InterORB (RMI/IIOP) w celu wywołania metody EJB na innym serwerze.



Rysunek 29. Zdalne wywołanie komponentu EJB

Podczas konfigurowania powiązania EJB produkt Integration Designer używa nazwy JNDI, aby określić poziom modelu programowania komponentu EJB oraz typ wywołania (lokalne lub zdalne).

Powiązania importu EJB zawierają następujące główne komponenty:

- procedurę obsługi danych JAX-WS,
- selektor błędu komponentu EJB,
- selektor funkcji importu EJB.

Jeśli scenariusz użytkownika nie jest oparty na odwzorowaniu JAX-WS, do wykonania czynności realizowanych w innych przypadkach przez komponenty stanowiące część powiązań importu EJB mogą być potrzebne niestandardowa procedura obsługi danych, selektor funkcji i selektor błędu. Dotyczy to również odwzorowania zazwyczaj realizowanego przez niestandardowy algorytm odwzorowania.

Powiązania eksportu EJB:

Zewnętrzne aplikacje Java EE mogą wywoływać komponent SCA za pośrednictwem powiązania eksportu EJB. Jeśli wykorzystywany jest eksport EJB, istnieje możliwość ujawniania komponentów SCA tak, aby zewnętrzne aplikacje Java EE mogły wywoływać te komponenty przy użyciu modelu programowania EJB.

Uwaga: eksport komponentu EJB jest bezstanowym komponentem bean.

Do tworzenia powiązań EJB służy produkt Integration Designer. Do wygenerowania powiązań można użyć jednej z poniższych procedur:

- Tworzenie powiązań eksportu EJB przy użyciu kreatora usług zewnętrznych
Do zbudowania usługi eksportu EJB na podstawie istniejącej implementacji można użyć kreatora usług zewnętrznych w produkcie Integration Designer. Kreator usług zewnętrznych umożliwia tworzenie usług na podstawie podawanych przez użytkownika kryteriów. Generuje on następnie obiekty biznesowe, interfejsy oraz pliki eksportu w oparciu o wykryte usługi.
- Tworzenie powiązań eksportu EJB przy użyciu edytora składania
Istnieje możliwość utworzenia eksportu EJB przy użyciu edytora składania produktu Integration Designer.

Ważne: klient J2SE (Java 2 Platform, Standard Edition) nie może wywoływać klienta eksportu EJB wygenerowanego w produkcie Integration Designer.

Użytkownik może wygenerować powiązanie z istniejącego komponentu SCA lub wygenerować eksport z powiązaniem EJB dla interfejsu Java.

- Jeśli użytkownik generuje eksport dla istniejącego komponentu SCA zawierającego istniejący interfejs WSDL, do eksportu przypisywany jest interfejs Java.
- Jeśli użytkownik generuje eksport dla interfejsu Java, można wybrać interfejs WSDL lub interfejs Java dla eksportu.

Uwaga: interfejs Java używany do tworzenia eksportu EJB charakteryzuje się następującymi ograniczeniami obiektów (parametrów wejściowych i wyjściowych oraz wyjątków), które są przekazywane jako parametry w wywołaniu zdalnym:

- Obiekty te muszą mieć konkretny typ (zamiast typu interfejsu lub typu abstrakcyjnego).
- Obiekty te muszą być zgodne ze specyfikacją EJB (Enterprise JavaBeans). Musi być możliwe ich serializowanie i muszą mieć domyślny konstruktor no-argument, zaś wszystkie ich właściwości muszą być dostępne za pośrednictwem metod procedur pobierających i procedur ustawiających.

Informacje na temat specyfikacji Enterprise JavaBeans zawiera serwis WWW firmy Sun Microsystems Inc. - <http://java.sun.com>.

Dodatkowo wyjątek musi być wyjątkiem sprawdzonym, dziedziczonym z klasy `java.lang.Exception` i musi być pojedynczy (czyli nie może obsługiwać zgłaszania wielu sprawdzonych typów wyjątków).

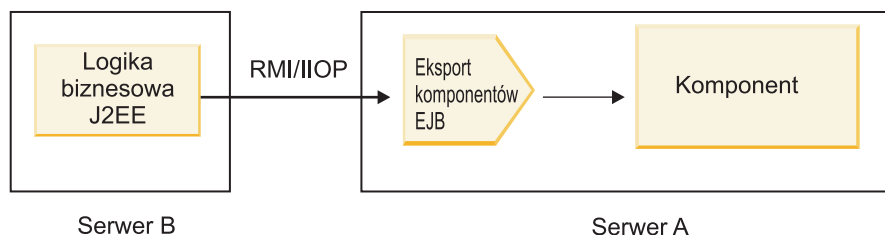
Należy również zauważyć, że interfejs biznesowy komponentu EJB języka Java jest zwykłym interfejsem Java i nie może rozszerzać interfejsu `javax.ejb.EJBObject` ani interfejsu `javax.ejb.EJBLocalObject`. Metody interfejsu biznesowego nie powinny zgłaszać wyjątku `java.rmi.Remote.Exception`.

Powiązania eksportu EJB mogą realizować interakcję z logiką biznesową Java EE przy użyciu modelu programowania EJB 2.1 lub modelu programowania EJB 3.0.

Wywołanie może być lokalne (tylko dla modelu EJB 3.0) lub zdalne.

- Wywołanie lokalne jest używane, jeśli logika biznesowa Java EE wywołuje komponent SCA znajdujący się na tym samym serwerze co eksport.
- Wywołanie zdalne jest używane, jeśli logika biznesowa Java EE nie znajduje się na tym samym serwerze co eksport.

Na przykład na poniższej ilustracji komponent EJB używa technologii RMI/IIOP do wywoływania komponentu SCA na innym serwerze.



Rysunek 30. Wywołanie zdalne z klienta do komponentu SCA za pośrednictwem eksportu EJB

Podczas konfigurowania powiązania EJB produkt Integration Designer używa nazwy JNDI, aby określić poziom modelu programowania komponentu EJB oraz typ wywołania (lokalne lub zdalne).

Powiązania eksportu EJB zawierają następujące główne komponenty:

- procedurę obsługi danych JAX-WS,
- selektor funkcji eksportu EJB.

Jeśli scenariusz użytkownika nie jest oparty na odwzorowaniu JAX-WS, do wykonania czynności realizowanych w innych przypadkach przez komponenty stanowiące część powiązań eksportu EJB mogą być potrzebne niestandardowa procedura obsługi danych i selektor funkcji. Dotyczy to również odwzorowania zazwyczaj realizowanego przez niestandardowy algorytm odwzorowania.

Właściwości powiązania EJB:

Powiązania importu EJB używają skonfigurowanych nazw JNDI do określenia poziomu modelu programowania komponentu EJB oraz typu wywołania (lokalne lub zdalne). Powiązania importu i eksportu EJB używają procedury obsługi danych JAX-WS do transformowania danych. Powiązanie importu EJB używa selektora funkcji importu EJB i selektora błędu EJB, a powiązanie eksportu EJB używa selektora funkcji eksportu EJB.

Nazwy JNDI i powiązania importu EJB:

Podczas konfigurowania powiązania EJB dla importu produkt Integration Designer używa nazwy JNDI, aby określić poziom modelu programowania komponentu EJB oraz typ wywołania (lokalne lub zdalne).

Jeśli nie zostanie określona nazwa JNDI, używane jest domyślne powiązanie interfejsu EJB. Tworzone nazwy domyślne zależą od tego, czy wywoływany jest komponent JavaBeans standardu EJB 2.1, czy komponent JavaBeans standardu EJB 3.0.

Uwaga: Więcej informacji na temat konwencji nazewnictwa zawiera temat „Przegląd powiązań aplikacji EJB 3.0” dostępny w Centrum informacyjnym produktu WebSphere Application Server.

- Komponenty JavaBean standardu EJB 2.1

Domyślna nazwa JNDI wstępnie wybrana przez produkt Integration Designer to domyślne powiązanie EJB 2.1, które przyjmuje postać łańcucha **ejb/** skonkatenowanego z interfejsem głównym rozdzielanym ukośnikami.

Na przykład w przypadku interfejsu głównego komponentu JavaBeans w standardzie EJB 2.1 `com.mojafirma.mojzdalnybiznes` domyślnym powiązaniem jest:

```
ejb/com/mojafirma/mojzdalnybiznes
```

W przypadku standardu EJB 2.1 obsługiwane jest tylko zdalne wywoływanie komponentów EJB.

- Komponenty JavaBeans standardu EJB 3.0

Domyślna nazwa JNDI wstępnie wybrana przez produkt Integration Designer dla lokalnej nazwy JNDI jest pełną nazwą klasy interfejsu lokalnego poprzedzoną łańcuchem **ejblocal:**. Na przykład wstępnie wybraną nazwą JNDI w standardzie EJB 3.0 dla pełnego interfejsu lokalnego `com.mojafirma.mojbizneslokalny` jest:

```
ejblocal:com.mojafirma.mojbizneslokalny
```

W przypadku zdalnego interfejsu `com.mojafirma.mojbizneszdalny` wybraną nazwą JNDI w standardzie EJB 3.0 jest pełny interfejs:

```
com.mojafirma.mojbizneszdalny
```

Domyślne powiązania aplikacji EJB 3.0 są opisane w następującym miejscu: EJB 3.0 application bindings overview (Przegląd powiązań aplikacji EJB 3.0).

Jako domyślnego położenia JNDI dla komponentów EJB korzystających z modelu programowania w wersji 3.0 produkt Integration Designer używa nazwy skróconej.

Uwaga: Jeśli z powodu zastosowania lub skonfigurowania niestandardowego odwzorowania wdrożone odwołanie JNDI dla docelowego komponentu EJB jest różne niż domyślne położenie powiązania JNDI, należy poprawnie określić docelową nazwę JNDI. Nazwę tę można określić w produkcie Integration Designer przed wdrożeniem, a w przypadku powiązania importu można ją również zmienić w Konsoli administracyjnej (po wdrożeniu) tak, aby była zgodna z nazwą JNDI docelowego komponentu EJB.

Więcej informacji na temat tworzenia powiązań EJB zawiera sekcja poświęcona pracy z powiązaniem EJB w Centrum informacyjnym produktu Integration Designer.

Procedura obsługi danych JAX-WS:

Powiązanie importu EJB (Enterprise JavaBeans) stosuje procedurę obsługi danych JAX-WS do przekształcania obiektów biznesowych żądania w parametry obiektu Java oraz do przekształcania wartości zwracanej obiektu Java w obiekt biznesowy odpowiedzi. Powiązanie eksportu EJB używa procedury obsługi danych JAX-WS do przekształcania komponentów EJB żądania w obiekty biznesowe żądania oraz do przekształcania obiektu biznesowego odpowiedzi w wartość zwracaną.

Ta procedura obsługi danych odwzorowuje dane z interfejsu WSDL określonego w architekturze SCA na docelowy interfejs Java komponentu EJB (i odwrotnie). Wykorzystuje w tym celu specyfikację JAX-WS (Java API for XML Web Services) i specyfikację JAXB (Java Architecture for XML Binding).

Uwaga: Aktualnie obsługa jest ograniczona do specyfikacji JAX-WS 2.1.1 i JAXB 2.1.3.

Procedura obsługi danych określona na poziomie powiązania EJB jest używana do przetwarzania żądań, odpowiedzi, błędów i wyjątków czasu wykonywania.

Uwaga: W przypadku błędów istnieje możliwość określenia specyficznej procedury obsługi danych dla każdego błędu poprzez określenie właściwości konfiguracji `faultBindingType`. Powoduje to nadpisanie wartości określonej na poziomie powiązania EJB.

Procedura obsługi danych JAX-WS jest używana domyślnie, jeśli powiązanie EJB ma interfejs WSDL. Tej procedury obsługi danych nie można używać do transformowania komunikatu SOAP reprezentującego wywołanie JAX-WS na obiekt danych.

Powiązanie importu EJB używa procedury obsługi danych do transformowania obiektu danych w tablicę `Object` języka Java (`Object[]`). Podczas komunikacji wychodzącej realizowane są następujące działania przetwarzania:

1. Powiązanie EJB ustawia oczekiwany typ, oczekiwany element i nazwę metody docelowej we właściwości `BindingContext` tak, aby były zgodne z danymi określonymi w pliku WSDL.
2. Powiązanie EJB wywołuje metodę transformowania dla obiektu danych, który wymaga transformowania danych.
3. Procedura obsługi danych zwraca tablicę `Object[]` reprezentującą parametry metody (w kolejności ich definiowania wewnątrz metody).
4. Powiązanie EJB używa tablicy `Object[]` do wywołania metody dla docelowego interfejsu EJB.

Powiązanie przygotowuje również tablicę `Object[]` tak, aby przetwarzać odpowiedź z wywołania komponentu EJB.

- Pierwszy element w tablicy `Object[]` jest wartością zwracaną z wywołania metody Java.
- Kolejne wartości reprezentują parametry wejściowe dla metody.

Jest to wymagane do obsługi typów parametrów `In/Out` i `Out`.

W przypadku parametrów typu `Out` wartości muszą być zwracane w obiekcie danych odpowiedzi.

Procedura obsługi danych przetwarza i transformuje wartości znalezione w tablicy `Object[]`, a następnie zwraca odpowiedź do obiektu danych.

Procedura obsługi danych obsługuje elementy `xs:AnyType`, `xs:AnySimpleType` i `xs:Any`, a także inne typy danych XSD. Aby włączyć obsługę elementu `xs:Any`, należy użyć wartości `@XmlElement (lax=true)` dla właściwości komponentu `JavaBeans` w kodzie Java. Zostało to przedstawione w poniższym przykładzie:

```
public class TestType {
    private Object[] object;

    @XmlElement (lax=true)
    public Object[] getObject() {
        return object;
    }

    public void setObject (Object[] object) {
        this.object=object;
    }
}
```

Ten kod powoduje, że obiekt właściwości w klasie `TestType` jest polem `xs:any`. Wartość klasy Java używanej w polu `xs:any` powinna mieć adnotację `@XmlElement`. Jeśli na przykład `Address` jest klasą Java używaną do wypełniania tablicy obiektów, klasa `Address` powinna mieć adnotację `@XmlElement`.

Uwaga: Aby dostosować odwzorowanie z typu XSD na typy Java zdefiniowane przez specyfikację JAX-WS, należy zmienić adnotacje JAXB tak, aby spełniały potrzeby biznesowe. Procedura obsługi danych JAX-WS obsługuje elementy `xs:any`, `xs:anyType`, `xs:anySimpleType`.

Procedura obsługi danych JAX-WS podlega następującym ograniczeniom:

- Procedura obsługi danych nie obejmuje obsługi adnotacji **@WebParam** atrybutu nagłówka.
- Przestrzeń nazw dla plików schematu obiektu biznesowego (plików XSD) nie obejmuje domyślnego odwzorowania z nazwy pakietu Java. Adnotacja **@XMLSchema** w klasie `package-info.java` również nie działa. Jedynym sposobem utworzenia definicji XSD z przestrzenią nazw jest użycie adnotacji **@XmlType** i **@XmlRootElement**. Adnotacja **@XmlRootElement** definiuje docelową przestrzeń nazw dla elementu globalnego w typach komponentów JavaBeans.
- Kreator importu EJB nie tworzy plików XSD dla klas niepowiązanych. Wersja 2.0 nie obsługuje adnotacji **@XmlSeeAlso**, a więc jeśli klasa potomna nie jest przywoływana bezpośrednio z klasy macierzystej, nie zostanie utworzona definicja XSD. Rozwiązaniem tego problemu jest uruchomienie programu narzędziowego SchemaGen dla takich klas potomnych.

Program SchemaGen jest programem narzędziowym wiersza komend (znajdującym się w katalogu `katalog_główny_instalacji_produktu_WPS/bin`), udostępnianym w celu tworzenia plików XSD dla określonego komponentu bean. Te pliki XSD należy ręcznie skopiować do modułu, aby rozwiązanie zadziałało.

Selektor błędów komponentu EJB:

Selektor błędów komponentu EJB określa, czy wywołanie komponentu EJB zakończyło się błędem, wyjątkiem czasu wykonywania czy pomyślną odpowiedzią.

Jeśli został wykryty błąd, selektor błędów komponentu EJB zwraca rodzimą nazwę błędu do środowiska wykonawczego powiązania, aby procedura obsługi danych JAX-WS mogła przekształcić obiekt wyjątku w obiekt biznesowy błąd.

W przypadku pomyślnej odpowiedzi (niebędącej błędem), powiązanie importu EJB składa tablicę obiektów Java (`Object[]`) w celu zwrócenia wartości.

- Pierwszy element w tablicy `Object[]` jest wartością zwracaną z wywołania metody Java.
- Kolejne wartości reprezentują parametry wejściowe dla metody.

Jest to wymagane do obsługi typów parametrów In/Out i Out.

W przypadku scenariuszy wyjątków, powiązanie składa tablicę `Object[]`, a jej pierwszy element reprezentuje wyjątek zgłoszony przez metodę.

Selektor błędów może zwrócić jedną spośród następujących wartości:

Tabela 35. Wartości zwracane

Typ	Wartość zwracana	Opis
Błąd	ResponseType.FAULT	Wartość zwracana, jeśli przekazana tablica <code>Object[]</code> zawiera obiekt wyjątku.
Wyjątek czasu wykonywania	ResponseType.RUNTIME	Wartość zwracana, jeśli obiekt wyjątku nie jest zgodny z żadnym spośród zadeklarowanych typów wyjątku w metodzie.
Normalna odpowiedź	ResponseType.RESPONSE	Wartość zwracana we wszystkich pozostałych przypadkach.

Jeśli selektor błędów zwróci wartość **ResponseType.FAULT**, zwracana jest rodzima nazwa błędu. Ta rodzima nazwa błędu jest używana przez powiązanie w celu określenia odpowiedniej nazwy błędu WSDL z modelu oraz wywołania poprawnej procedury obsługi danych błędu.

Selektor funkcji komponentu EJB:

Powiązania EJB używają selektora funkcji importu (przy przetwarzaniu danych wychodzących) lub selektora funkcji eksportu (przy przetwarzaniu danych przychodzących), aby określić, która metoda EJB ma zostać wywołana.

Selektor funkcji importu

W przypadku przetwarzania danych wychodzących selektor funkcji importu określa typ metody EJB na podstawie nazwy operacji wywołanej przez komponent SCA połączony z importem EJB. Selektor funkcji wyszukuje adnotację `@WebMethod` w wygenerowanej przez produkt Integration Designer klasie Java z adnotacją JAX-WS, aby określić nazwę powiązanej operacji docelowej.

- Jeśli występuje adnotacja `@WebMethod`, selektor funkcji używa adnotacji `@WebMethod` do określenia poprawnego odwzorowania metody Java dla metody WSDL.
- Jeśli nie występuje adnotacja `@WebMethod`, selektor funkcji zakłada, że nazwa metody Java jest taka sama jak nazwa wywołanej operacji.

Uwaga: Ten selektor funkcji działa tylko w przypadku interfejsu typu WSDL w imporcie EJB, ale nie działa w przypadku interfejsu typu Java w imporcie EJB.

Selektor funkcji zwraca obiekt `java.lang.reflect.Method` reprezentujący metodę interfejsu EJB.

Selektor funkcji używa tablicy `Object` języka Java (`Object[]`), w której zawierana jest odpowiedź z metody docelowej. Pierwszy element tablicy `Object[]` jest metodą Java o nazwie pliku WSDL, a drugi element w tablicy `Object[]` jest wejściowym obiektem biznesowym.

Selektor funkcji eksportu

W przypadku przetwarzania danych wychodzących selektor funkcji eksportu określa, która metoda docelowa ma zostać wykonana, na podstawie metody Java.

Selektor funkcji eksportu odwzorowuje nazwę operacji Java wywołanej przez klienta EJB na nazwę operacji w interfejsie komponentu docelowego. Nazwa metody jest zwracana w postaci łańcucha i rozstrzygana przez środowisko wykonawcze SCA w zależności od typu interfejsu komponentu docelowego.

Powiązania EIS:

Powiązania EIS (Enterprise Information System) udostępniają połączenia między komponentami SCA i zewnętrznym systemem EIS. Ta komunikacja jest realizowana za pomocą eksportów EIS i importów EIS, które obsługują adaptery zasobów architektury JCA 1.5 i adaptery WebSphere Adapters.

Komponenty SCA mogą wymagać przesyłania danych do zewnętrznego systemu EIS lub z takiego systemu. Podczas tworzenia modułu SCA wymagającego takich połączeń należy dołączyć (oprócz komponentu SCA) import lub eksport z powiązaniem EIS umożliwiającym komunikowanie z określonym zewnętrznym systemem EIS.

Adaptory zasobów w produkcie IBM Integration Designer są używane w kontekście importu lub eksportu. Importy i eksporty są opracowywane za pomocą kreatora usług zewnętrznych, a w trakcie ich opracowywania dołączany jest adapter zasobu. Import EIS, który umożliwia aplikacji wywoływanie usługi w systemie EIS, lub eksport EIS, który umożliwia aplikacji w systemie EIS wywoływanie usługi utworzonej w produkcie IBM Integration Designer, są tworzone przy użyciu adaptera zasobu. Na przykład w celu wywołania usługi w systemie JD Edwards należy utworzyć import z adapterem systemu JD Edwards.

W przypadku używania kreatora usług zewnętrznych informacje powiązania EIS są tworzone dla użytkownika. Do dodawania lub modyfikowania informacji o powiązaniu można również użyć innego narzędzia - edytora składania. Więcej informacji zawiera sekcja Uzyskiwanie dostępu do usług zewnętrznych za pomocą adapterów.

Po wdrożeniu modułu SCA zawierającego powiązanie EIS można użyć Konsoli administracyjnej, aby wyświetlić informacje o powiązaniu lub aby je skonfigurować.

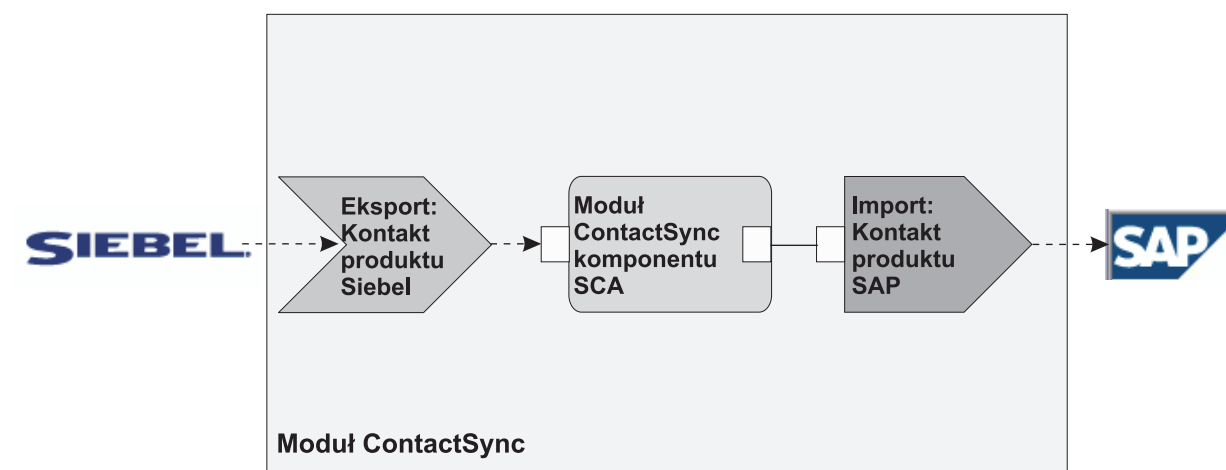
Przegląd powiązań EIS:

Powiązanie EIS (Enterprise Information System) używane z adapterem zasobu JCA umożliwia dostęp do usług w systemie EIS lub udostępnienie usług systemowi EIS.

Poniższy przykład przedstawia sposób, w jaki moduł SCA o nazwie ContactSyncModule synchronizuje informacje kontaktowe między systemem Siebel i systemem SAP.

1. Komponent SCA o nazwie ContactSync nasłuchuje (za pośrednictwem eksportu aplikacji EIS o nazwie Siebel Contact) zmian w kontaktach systemu Siebel.
2. Komponent SCA ContactSync korzysta z aplikacji SAP (za pośrednictwem importu aplikacji EIS), aby zaktualizować w odpowiedni sposób informacje kontaktowe systemu SAP.

Ponieważ struktury danych używane do przechowywania kontaktów są różne w systemach Siebel i SAP, komponent SCA ContactSync musi udostępniać odwzorowanie.



Rysunek 31. Przepływ z systemu Siebel do systemu SAP

Eksport kontaktów systemu Siebel i import kontaktów systemu SAP mają skonfigurowane odpowiednie adaptory zasobów.

Kluczowe opcje powiązań z systemami EIS:

Element importowany EIS jest elementem importowanym architektury Service Component Architecture (SCA) umożliwiającym komponentom zawartym w module SCA na korzystanie z aplikacji EIS zdefiniowanych poza modulem SCA. Element importowany EIS jest używany do przesyłania danych z komponentu SCA do zewnętrznego systemu EIS, natomiast element eksportowany EIS jest używany do przesyłania danych z zewnętrznego systemu EIS do modułu SCA.

Elementy importowane

Element importowany EIS służy do łączenia komponentów architektury SCA z zewnętrznymi systemami EIS. Aplikacje zewnętrzne można traktować jak element importowany EIS. W tym przypadku element importowany EIS wysyła dane do zewnętrznego systemu EIS i opcjonalnie odbiera dane wysyłane do niego w odpowiedzi.

Element importowany EIS udostępnia komponentom SCA jednolity widok aplikacji zewnętrznych w stosunku do modułu. Pozwala to komponentom komunikować się z zewnętrznym systemem EIS, takim jak SAP, Siebel lub PeopleSoft, z użyciem spójnego modelu SCA.

Po stronie klienta elementu importowanego istnieje interfejs ujawniony przez aplikację importu EIS zawierający jedną lub kilka metod, z których każda pobiera obiekty danych jako argumenty i zwraca wartości. Po stronie implementacji występuje interfejs CCI (Common Client Interface) zaimplementowany przez adapter zasobów.

Implementacja wykonawcza importu EIS łączy interfejs po stronie klienta z interfejsem CCI. Import odwzorowuje wywołanie metody interfejsu w wywołanie po stronie CCI.

Powiązania są tworzone na trzech poziomach: powiązanie interfejsów korzystające z zawartych w nich powiązań metod, w których używane są powiązania danych.

Powiązanie interfejsów łączy interfejs elementu importowanego z połączeniem z systemem EIS udostępniającym aplikację. Odzwierciedla to fakt, że zestaw aplikacji reprezentowany przez interfejs jest udostępniany przez określoną instancję systemu EIS, a połączenie zapewnia dostęp do tej instancji. Element wiążący obejmuje właściwości udostępniające informacje, które są wystarczające do utworzenia połączenia (te właściwości są częścią instancji interfejsu `javax.resource.spi.ManagedConnectionFactory`).

Powiązanie metod łączy metodę z określoną interakcją z systemem EIS. W przypadku architektury JCA interakcja jest charakteryzowana przez zestaw właściwości implementacji interfejsu `javax.resource.cci.InteractionSpec`. Element interakcji powiązania metod obejmuje właściwości oraz nazwę klasy. Informacje te są wystarczające do wykonania interakcji. Powiązanie metod korzysta z powiązań danych opisujących odwzorowanie argumentu i wyniku metody interfejsu w reprezentację w systemie EIS.

Scenariusz wykonawczy związany z elementem importowanym EIS jest następujący:

1. Metoda interfejsu importu jest wywoływana z użyciem modelu programowania architektury SCA.
2. Żądanie docierające do importu EIS zawiera nazwę metody i jej argumenty.
3. Import tworzy najpierw implementację powiązania interfejsów, a następnie, korzystając z danych pochodzących z powiązania elementu importowanego, tworzy interfejs `ConnectionFactory` i wiąże te dwa elementy. Innymi słowy import wywołuje interfejs `setConnectionFactory` w powiązaniu interfejsów.
4. Utworzona zostaje implementacja powiązania metod odpowiadająca wywołanej metodzie.
5. Instancja interfejsu `javax.resource.cci.InteractionSpec` zostaje utworzona i zapełniona, a następnie z użyciem powiązań danych argumenty metody zostają przekształcone w format zrozumiały dla adaptera zasobów.
6. Za pomocą interfejsu CCI przeprowadzona zostaje interakcja.
7. Podczas powrotu z wywołania z użyciem powiązania danych utworzony zostaje wynik wywołania, a następnie wynik ten jest zwracany do programu wywołującego.

Eksperty

Ekspert EIS służy do łączenia komponentu architektury SCA z zewnętrznym systemem EIS. Aplikacje zewnętrzne można traktować jak element eksportowany EIS. W tym przypadku zewnętrzna aplikacja wysyła swoje dane w postaci okresowych powiadomień. Element eksportowany EIS można uważać za aplikację subskrypcji nasłuchującą zewnętrznych żądań z systemu EIS. Komponent SCA korzystający z elementu eksportowanego EIS widzi go jako aplikację lokalną.

Element eksportowany EIS udostępnia komponentom SCA jednolity widok aplikacji zewnętrznych w stosunku do modułu. Pozwala to komponentom komunikować się z systemem EIS, takim jak SAP, Siebel lub PeopleSoft, z użyciem spójnego modelu SCA.

Element eksportowany obejmuje implementację procesu nasłuchiwanie odbierającego żądania z systemu EIS. Proces nasłuchiwanie implementuje interfejs procesu nasłuchiwanie specyficznego dla danego adaptera zasobów. Element eksportowany obejmuje także komponent implementujący interfejs ujawniony systemowi EIS za pośrednictwem tego elementu eksportowanego.

Implementacja wykonawcza elementu eksportowanego EIS łączy proces nasłuchiwanie z komponentem implementującym interfejs. Element eksportowany odwzorowuje żądanie systemu EIS w wywołanie odpowiedniej

operacji w komponencie. Powiązania są tworzone na trzech poziomach: powiązanie procesu nasłuchiwania korzystające z zawartego w nim powiązania metod własnych, w którym używane jest powiązanie danych.

Powiązanie procesu nasłuchiwania łączy proces nasłuchiwania odbierający żądania z komponentem ujawnionym za pośrednictwem elementu eksportowanego. Definicja elementu eksportowanego obejmuje nazwę komponentu. Środowisko wykonawcze znajduje ten komponent, a następnie przekazuje do niego żądania.

Powiązanie metod własnych wiąże metodę własną lub typ zdarzenia odebranego przez proces nasłuchiwania z operacją implementowaną przez komponent ujawniony za pomocą elementu eksportowanego. Nie ma żadnego związku między metodą wywołaną w procesie nasłuchiwania i typem zdarzenia. Wszystkie zdarzenia nadchodzą za pośrednictwem jednej lub kilku metod procesu nasłuchiwania. Powiązanie metod własnych korzysta z selektora funkcji zdefiniowanego w elemencie eksportowanym w celu wyodrębnienia nazwy metody własnej z przychodzących danych, a powiązań danych w celu powiązania formatu danych systemu EIS z formatem zrozumiałym dla komponentu.

Scenariusz wykonawczy związany z elementem eksportowanym EIS jest następujący:

1. Żądanie systemu EIS powoduje wywołanie metody w implementacji procesu nasłuchiwania.
2. Proces nasłuchiwania odnajduje i wywołuje element eksportowany i przekazuje do niego wszystkie argumenty wywołania.
3. Element eksportowany tworzy implementację powiązania procesu nasłuchiwania.
4. Element eksportowany tworzy instancję selektora funkcji i ustawia go na powiązanie procesu nasłuchiwania.
5. Element eksportowany inicjuje powiązania metod własnych i dodaje je do powiązania procesu nasłuchiwania. Dla każdego powiązania metod własnych zostają także zainicjowane powiązania danych.
6. Element eksportowany wywołuje powiązanie procesu nasłuchiwania.
7. Powiązanie procesu nasłuchiwania odnajduje komponenty i używa selektora funkcji do pobrania nazwy metody własnej.
8. Ta nazwa umożliwia odnalezienie powiązania metod własnych, które następnie wywołuje komponent docelowy.

Styl interakcji adaptera pozwala powiązaniu elementu eksportowanego EIS wywołać komponent docelowy asynchronicznie (opcja domyślna) lub synchronicznie.

Adaptory zasobów

Elementy importowane i eksportowane są opracowywane za pomocą kreatora usługi zewnętrznej, a w trakcie ich opracowywania dołączany jest adapter zasobów. Adaptory dostarczane z produktem IBM Integration Designer zapewniające dostęp do systemów CICS, IMS, JD Edwards, PeopleSoft, SAP i Siebel są przeznaczone wyłącznie do programowania i testowania. Oznacza to, że są one używane tylko podczas programowania i testowania aplikacji.

Po wdrożeniu aplikacji do jej uruchomienia wymagane są licencjonowane adaptory wykonawcze. Jednak podczas budowania usługi adapter można w niej osadzić. Licencja adaptera może pozwalać użyć adaptera osadzonego jako licencjonowanego adaptera wykonawczego. Adaptory te są zgodne z architekturą Java EE Connector Architecture (JCA 1.5). Architektura JCA jest otwartym standardem platformy Java EE zapewniającym łączność z systemem EIS. Architektura JCA udostępnia środowisko zarządzane, co oznacza, że serwer aplikacji obsługuje mechanizm Quality of Service (QoS) zapewniający zarządzanie cyklem życia oraz bezpieczeństwo transakcji. Są one także zgodne ze specyfikacją Enterprise Metadata Discovery z wyjątkiem modułów IBM CICS ECI Resource Adapter i IBM IMS Connector for Java.

Kreator obsługuje także starszy zestaw adapterów, czyli adaptory produktu WebSphere Business Integration.

Zasoby środowiska Java EE

Moduł EIS, czyli moduł SCA zgodny ze wzorcem modułów EIS, można wdrożyć na platformie Java EE.

Wynikiem wdrożenia modułu EIS na platformie Java EE jest gotowa do uruchomienia aplikacja spakowana do postaci pliku EAR i wdrożona na serwerze. Wszystkie artefakty i zasoby platformy Java EE zostają utworzone, a aplikacja jest skonfigurowana i gotowa do uruchomienia.

Właściwości dynamiczne JCA specyfikacji interakcji i specyfikacji połączenia:

Powiązanie EIS może akceptować dane wejściowe dla interfejsów InteractionSpec i ConnectionSpec określone przy użyciu precyzyjnie zdefiniowanego obiektu danych potomnych dołączonego do ładunku. Umożliwia to realizowanie dynamicznych interakcji żądanie-odpowiedź z adapterem zasobu za pośrednictwem interfejsu InteractionSpec oraz uwierzytelnianie komponentów za pośrednictwem interfejsu ConnectionSpec.

Interfejs javax.cci.InteractionSpec przesyła informacje o sposobie obsługi żądania interakcji z adapterem zasobu. Może również przesyłać informacje na temat tego, w jaki sposób uzyskana została interakcja po wykonaniu żądania. Ten sposób dwukierunkowej komunikacji za pośrednictwem interakcji jest czasem nazywany *konwersacjami*.

Powiązanie EIS oczekuje, aby ładunek będący argumentem adaptera zasobu zawierał obiekt danych potomnych o nazwie **properties**. Ten obiekt danych właściwości będzie zawierał pary nazwa-wartość zawierające nazwę właściwości specyfikacji interakcji w określonym formacie. Stosowane są następujące reguły formatowania:

- Nazwy muszą rozpoczynać się przedrostkiem **IS**, po którym powinna występować nazwa właściwości. Na przykład specyfikacja interakcji z właściwością JavaBeans o nazwie **InteractionId** powinna określać nazwę właściwości jako **ISInteractionId**.
- Para nazwa-wartość reprezentuje nazwę i wartość prostego typu właściwości specyfikacji interakcji.

Interfejs przedstawiony w tym przykładzie określa, że danymi wejściowymi operacji jest obiekt danych **Account**. Ten interfejs wywołuje aplikację powiązania importu EIS w celu wysyłania i odbierania dynamicznej właściwości interfejsu InteractionSpec o nazwie **workingSet** i wartości **xyz**.

Graf biznesowy lub obiekty biznesowe na serwerze zawierają bazowy obiekt biznesowy **properties**, który umożliwia wysyłanie danych specyficznych dla protokołu wraz z ładunkiem. Ten obiekt biznesowy **properties** jest wbudowany i nie trzeba go określać w schemacie XML podczas konstruowania obiektu biznesowego. Wymaga on jedynie utworzenia i użycia. Jeśli użytkownik ma własne typy danych zdefiniowane na podstawie schematu XML, należy określić element **properties** zawierający oczekiwane pary nazwa-wartość.

```
BOfactory dataFactory = (BOfactory) \
serviceManager.locateService("com/ibm/websphere/bo/BOfactory");
//Opakowanie dla interfejsów w stylu document/literal wrapped
//przejście do ładunku w przypadku elementów innych niż document/literal
DataObject docLitWrapper = dataFactory.createElement /
("http://mytest/eis/Account", "AccountWrapper");
```

Utworzenie ładunku.

```
DataObject account = docLitWrapper.createDataObject(0);
DataObject accountInfo = account.createDataObject("AccountInfo");
//Konfigurowanie ładunku
```

```
//Konstruowanie danych właściwości na potrzeby interakcji dynamicznej
```

```
DataObject properties = account.createDataObject("properties");
```

Określanie oczekiwanej wartości (**xyz**) dla nazwy **workingSet**.

```
properties.setString("ISworkingSet", "xyz");
```

```
//Wywołanie usługi z argumentem
```

```
Service accountImport = (Service) \
serviceManager.locateService("AccountOutbound");
DataObject result = accountImport.invoke("createAccount", docLitWrapper);
```

```
//Uzyskanie zwróconej właściwości
DataObject retProperties = result.getDataObject("properties");

String workingset = retProperties.getString("ISworkingSet");
```

Użytkownik może użyć właściwości interfejsu ConnectionSpec w celu dynamicznego uwierzytelnienia komponentu. Mają zastosowanie te same reguły co powyżej, oprócz tego, że przedrostkiem nazwy właściwości musi być **CS** (zamiast **IS**). Właściwości interfejsu ConnectionSpec nie są dwukierunkowe. Ten sam obiekt danych **properties** może zawierać zarówno właściwości IS, jak i CS.

Aby użyć właściwości interfejsu ConnectionSpec, należy ustawić atrybut **resAuth** określany w powiązaniu importu na wartość **Application**. Należy również upewnić się, czy adapter zasobu obsługuje autoryzację komponentu. Szczegółowe informacje zawiera rozdział 8 specyfikacji J2EE Connector Architecture Specification (specyfikacji architektury konektora J2EE).

Klienci zewnętrzne z powiązaniem EIS:

Serwer może wysyłać komunikaty do klientów zewnętrznych i odbierać komunikaty od nich za pośrednictwem powiązań EIS.

Klient zewnętrzny, na przykład portal WWW lub system EIS, może wymagać wysyłania komunikatów do modułu SCA na serwerze lub może wymagać wywołania przez komponent znajdujący się na serwerze.

Klient wywołuje import EIS w taki sam sposób, jak w przypadku innych aplikacji, przy użyciu interfejsu wywoływania dynamicznego (DII) lub interfejsu Java.

1. Klient zewnętrzny tworzy instancję klasy ServiceManager i wyszukuje import EIS przy użyciu jej nazwy odwołania. Wynikiem wyszukiwania jest implementacja interfejsu usługi.
2. Klient tworzy argument wejściowy - ogólny obiekt danych utworzony dynamicznie przy użyciu schematu obiektu danych. Ten krok jest realizowany przy użyciu implementacji interfejsu DataFactory obiektu SDO.
3. Klient zewnętrzny wywołuje system EIS i uzyskuje wymagane wyniki.

Klient może też wywołać import EIS przy użyciu interfejsu Java.

1. Klient tworzy instancję klasy ServiceManager i wyszukuje import EIS przy użyciu jej nazwy odwołania. Wynikiem wyszukiwania jest interfejs Java importu EIS.
2. Klient tworzy argument wejściowy i obiekt danych określonego typu.
3. Klient wywołuje system EIS i uzyskuje wymagane wyniki.

Interfejs eksportu EIS definiuje interfejs wyeksportowanego komponentu SCA, który jest dostępny dla zewnętrznych aplikacji EIS. Ten interfejs może być postrzegany jako interfejs wywoływany przez aplikację zewnętrzną (na przykład SAP lub PeopleSoft) za pośrednictwem implementacji środowiska wykonawczego aplikacji eksportu EIS.

Eksport używa klasy EISExportBinding do powiązania wyeksportowanych usług z zewnętrzną aplikacją EIS. Umożliwia to subskrybowanie aplikacji zawartej w module SCA w celu nasłuchiwanie żądań usługi EIS. Powiązanie eksportu EIS określa odwzorowanie między definicją zdarzeń przychodzących, w sposób postrzegany przez adapter zasobu (przy użyciu interfejsów architektury Java EE Connector Architecture - JCA), a wywołaniem operacji SCA.

Klasa EISExportBinding wymaga, aby zewnętrzne usługi EIS były oparte na kontraktach przychodzących architektury Java EE Connector Architecture (JCA) 1.5. Klasa EISExportBinding wymaga też, aby na poziomie powiązania lub na poziomie metody określone były procedura obsługi danych lub powiązanie danych.

Powiązania JMS:

Dostawca JMS (Java Message Service) umożliwia przesyłanie komunikatów na podstawie interfejsu API i modelu programowania JMS (Java Message Service). Udostępnia on fabryki połączeń JMS, które umożliwiają tworzenie połączeń dla miejsc docelowych JMS oraz wysyłanie i odbieranie komunikatów.

Powiązania JMS można użyć w interakcjach z powiązaniem dostawcy magistrali integracji usług SIB (Service Integration Bus). Są one zgodne ze specyfikacjami JMS i JCA 1.5.

Powiązania eksportu i importu JMS umożliwiają modułowi SCA (Service Component Architecture) wywoływanie zewnętrznych systemów JMS i odbieranie komunikatów z tych systemów.

Powiązania importu i eksportu JMS udostępniają integrację z aplikacjami JMS za pośrednictwem dostawcy JMS magistrali SIB opartego na architekturze JCA 1.5, który stanowi część serwera WebSphere Application Server. Inne adaptory zasobów JMS oparte na architekturze JCA 1.5 nie są obsługiwane.

Przegląd powiązań JMS:

Powiązania JMS udostępniają połączenia między środowiskiem architektury SCA (Service Component Architecture) i systemami JMS.

Powiązania JMS

Główne komponenty powiązań importu JMS i powiązań eksportu JMS są następujące:

- Adapter zasobów: umożliwia zarządzane, dwukierunkowe połączenia między modułem SCA i zewnętrznymi systemami JMS.
- Połączenia: hermetyzują wirtualne połączenie między klientem i aplikacją dostawcy.
- Miejsca docelowe: używane przez klienta do określenia celu tworzonych komunikatów lub źródła konsumowanych komunikatów.
- Dane uwierzytelniania: używane w celu zabezpieczenia dostępu do powiązania.

Kluczowe opcje powiązań usług JMS

Nagłówki specjalne

W elementach importowanych i eksportowanych JMS stosowane są specjalne właściwości nagłówka zawierające informacje pozwalające elementowi docelowemu na właściwą obsługę komunikatu.

Na przykład właściwość `TargetFunctionName` odwzorowuje metodę własną w metodę operacji.

Zasoby środowiska Java EE

Podczas wdrażania elementów importowanych i eksportowanych JMS w środowisku Java EE utworzona zostaje pewna liczba zasobów środowiska Java EE.

Fabryka połączeń

Używana przez klienty do tworzenia połączenia z dostawcą usług JMS.

Specyfikacja aktywacji

Używana przez elementy importowane do odbierania odpowiedzi na żądanie, a przez elementy eksportowane podczas konfigurowania punktów końcowych komunikatów reprezentujących procesy nasłuchiwanie komunikatów w ich interakcjach z systemem przesyłania komunikatów.

Miejsca docelowe

- Miejsca docelowe wysyłania: w elemencie importowanym jest to miejsce, do którego jest wysyłane żądanie lub komunikat wychodzący; w elemencie eksportowanym jest to miejsce, do którego jest wysyłany komunikat odpowiedzi, jeśli nie jest zastąpione za pomocą pola nagłówka `JMSReplyTo` w komunikacie przychodzącym.

- Miejsce docelowe odbioru: jest to miejsce, w którym należy umieścić komunikat przychodzący. W przypadku elementów importowanych jest to odpowiedź, a w przypadku elementów eksportowanych - żądanie.
- Miejsce docelowe wywołania zwrotnego: jest to miejsce docelowe w systemie JMS architektury SCA używane do zapisywania informacji o korelacji. Tego miejsca docelowego nie należy odczytywać ani nie należy do niego zapisywać.

Czynność instalacji powoduje utworzenie zasobu fabryki połączeń oraz trzech miejsc docelowych. Jej wynikiem jest także utworzenie zasobu specyfikacji aktywowania umożliwiającego procesowi nasłuchiwania komunikatów czasu wykonywania nasłuchiwanie odpowiedzi w miejscu docelowym odbioru. Właściwości tych zasobów są określone w pliku importu lub eksportu.

Integracja JMS i adaptery zasobów:

Interfejs JMS (Java Message Service) udostępnia integrację za pośrednictwem dostępnego adaptera zasobu JMS opartego na architekturze JCA 1.5. Dla adaptera zasobu JMS magistrali integracji usług (Service Integration Bus - SIB) udostępniana jest pełna obsługa integracji JMS.

Aby przeprowadzić integrację z zewnętrznym systemem JMS zgodnym ze standardem JCA 1.5, należy użyć adaptera zasobu dostawcy JMS dla architektury JCA 1.5. Usługi zewnętrzne zgodne ze standardem JCA 1.5 mogą odbierać komunikaty i wysyłać komunikaty w celu integracji z komponentami SCA (Service Component Architecture) przy użyciu adaptera zasobu JMS magistrali SIB.

Używanie innych adapterów zasobów architektury JCA 1.5 pochodzących od konkretnych dostawców nie jest obsługiwane.

Powiązania importu i eksportu JMS:

Powiązania importu i eksportu JMS umożliwiają interakcje modułów SCA z usługami udostępnianymi przez zewnętrzne aplikacje JMS.

Powiązania importu JMS

Połączenia z powiązaniem dostawcą JMS miejsc docelowych JMS są tworzone przy użyciu fabryki połączeń JMS. Do zarządzania fabrykami połączeń JMS dla domyślnego dostawcy przesyłania komunikatów należy użyć obiektów administracyjnych fabryki połączeń.

Interakcja z zewnętrznymi systemami JMS obejmuje używanie miejsc docelowych do wysyłania żądań i odbierania odpowiedzi.

Obsługiwane są dwa typy scenariuszy używania dla powiązania importu JMS, zależne od typu wywoływanej operacji:

- Jednokierunkowy: import JMS umieszcza komunikat w miejscu docelowym wysyłania skonfigurowanym w powiązaniu importu. Pole replyTo nagłówka JMS nie jest ustawiane.
- Dwukierunkowy (żądanie-odpowiedź): import JMS umieszcza komunikat w miejscu docelowym wysyłania, a następnie utrzymuje odpowiedź otrzymaną z komponentu SCA.

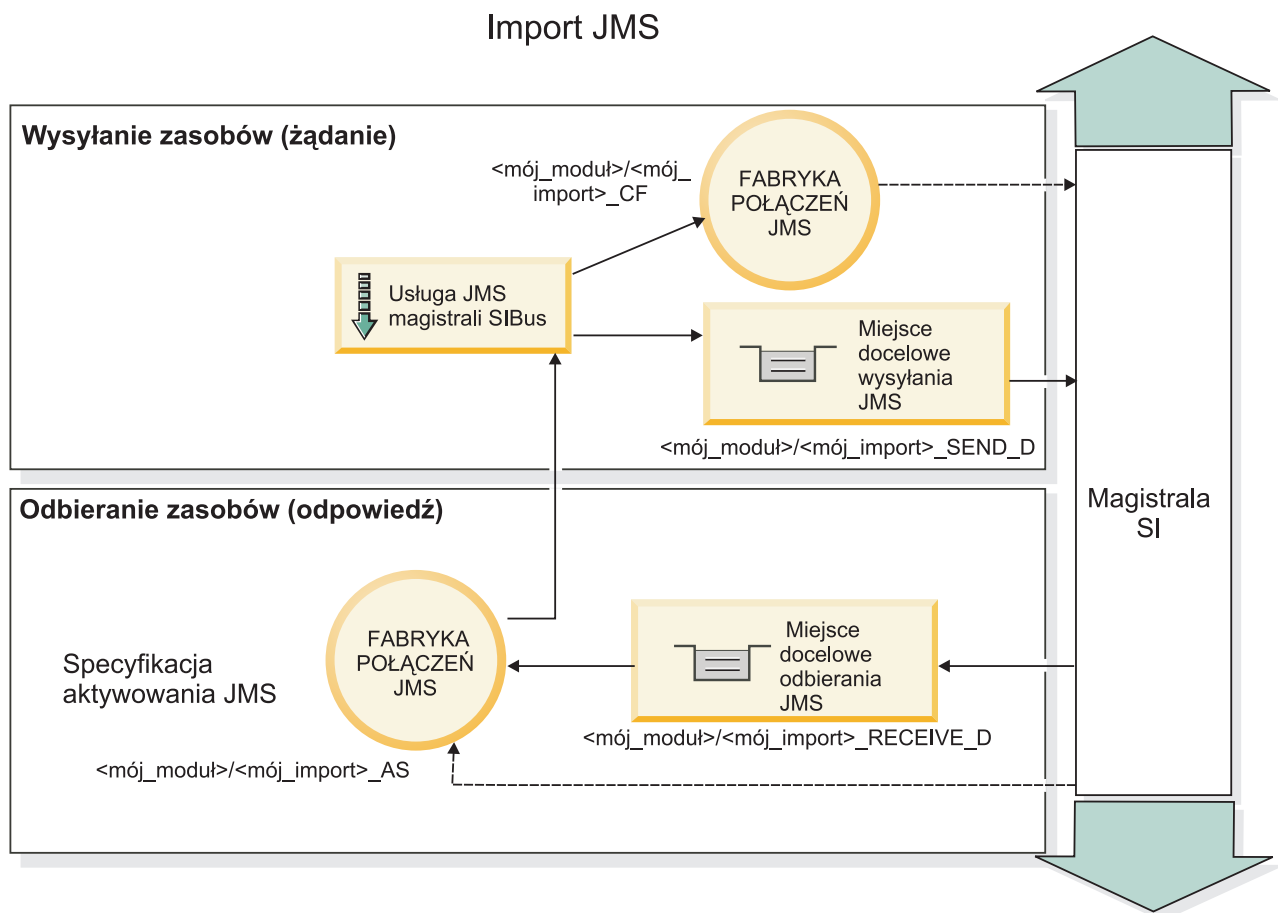
Powiązanie importu może być skonfigurowane (przy użyciu pola **Schemat korelacji odpowiedzi** w produkcie Integration Designer) tak, aby oczekiwać skopiowania identyfikatora korelacji komunikatu odpowiedzi z identyfikatora komunikatu żądania (wartość domyślna) lub z identyfikatora korelacji komunikatu żądania. Powiązanie importu może również być skonfigurowane tak, aby używać tymczasowego, dynamicznego miejsca docelowego odpowiedzi w celu korelowania odpowiedzi z żądaniami. Tymczasowe miejsce docelowe jest tworzone dla każdego żądania, a import używa tego miejsca docelowego do odbierania odpowiedzi.

Miejsce docelowe receive (odbioru) jest ustawiane we właściwości nagłówka replyTo komunikatu wychodzącego. Wdrażany jest proces nasłuchiwania komunikatów, który nasłuchuje w miejscu docelowym odpowiedzi, a po otrzymaniu odpowiedzi ten proces nasłuchiwania komunikatów przekazuje odpowiedź z powrotem do komponentu.

Istnieje możliwość określenia dynamicznych i statycznych właściwości nagłówka zarówno dla jednokierunkowych, jak i dwukierunkowych scenariuszy używania. Właściwości statyczne można ustawić z poziomu powiązania metody importu JMS. Niektóre z tych właściwości mają specjalne znaczenie dla środowiska wykonawczego JMS architektury SCA.

Należy pamiętać, że powiązanie JMS jest powiązaniem asynchronicznym. Jeśli komponent wywołujący wywołuje import JMS w sposób synchroniczny (dla operacji dwukierunkowej), komponent wywołujący jest blokowany do chwili zwrócenia odpowiedzi przez usługę JMS.

Na ilustracji Rys. 32 przedstawiono sposób połączenia importu z usługą zewnętrzną.



Rysunek 32. Zasoby powiązania importu JMS

Powiązania eksportu JMS

Powiązania eksportu JMS zapewniają modułom SCA sposób na udostępnianie usług zewnętrznym aplikacjom JMS.

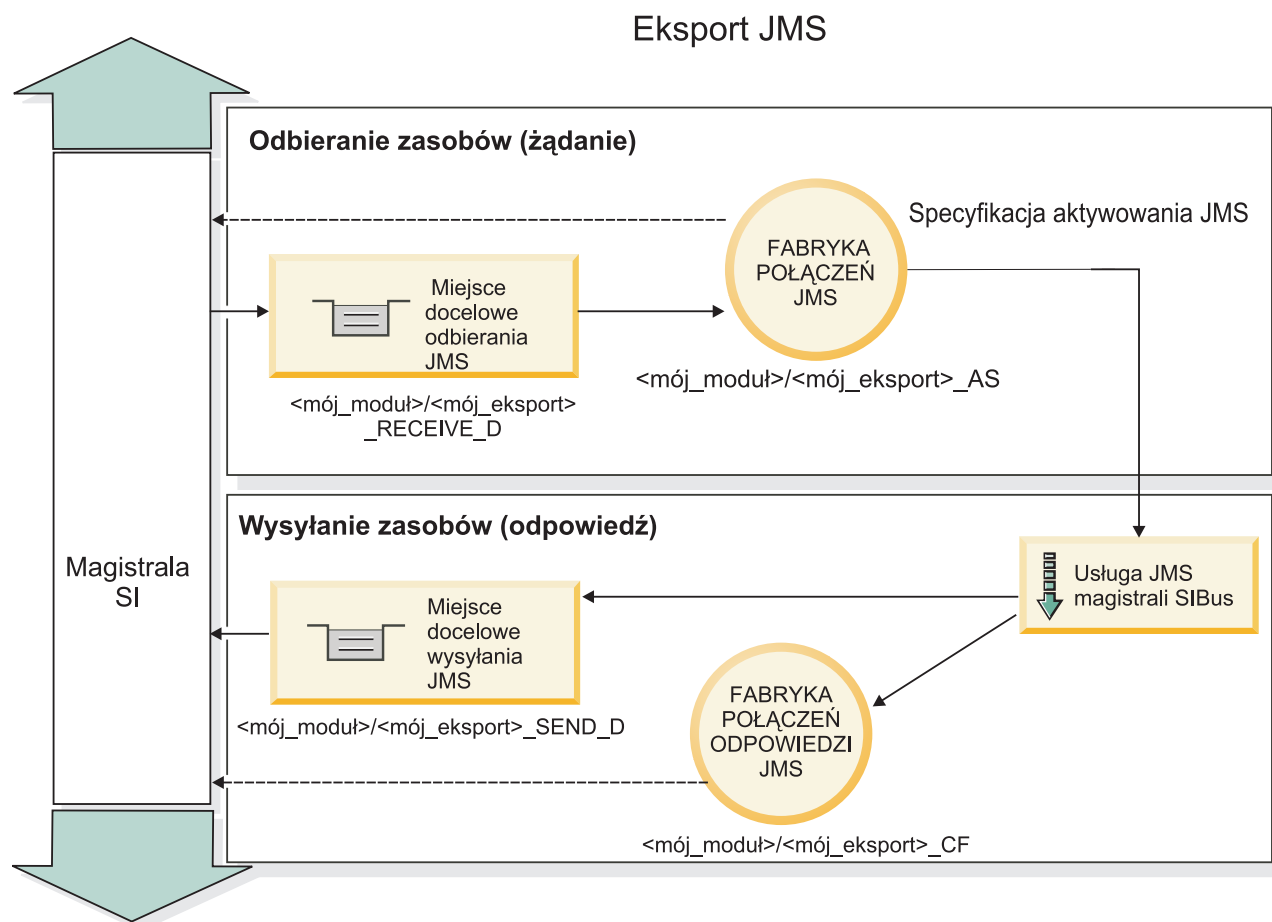
Powiązanie będące częścią eksportu JMS jest specyfikacją aktywowania, którą można konfigurować.

Eksport JMS ma miejsca docelowe wysyłania i odbierania.

- Miejsce docelowe receive (odbioru) jest miejscem, w którym powinny być umieszczane komunikaty przychodzące do komponentu docelowego.
- Miejsce docelowe send (wysyłania) jest miejscem, do którego wysyłana będzie odpowiedź, chyba że komunikat przychodzący nadpisuje to miejsce przy użyciu właściwości nagłówka replyTo.

Wdrażany jest proces nasłuchiwania komunikatów, który nasłuchuje żądań przychodzących do miejsca docelowego odbierania określonego w powiązaniu eksportu. Miejsce docelowe określone w polu send służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołany komponent udostępni odpowiedź. Miejsce docelowe określone w polu replyTo komunikatu przychodzącego nadpisuje miejsce docelowe określone w polu send.

Na ilustracji Rys. 33 przedstawiono sposób połączenia zewnętrznego requestera z eksportem.



Rysunek 33. Zasoby powiązania eksportu JMS

Nagłówki JMS:

Komunikat JMS zawiera dwa typy nagłówków: nagłówek systemowy JMS i wiele właściwości JMS. Dostęp do obu typów nagłówków można uzyskać w module mediacji, w obiekcie SMO lub przy użyciu funkcji API ContextService.

Nagłówek systemu JMS

Nagłówek systemu JMS jest reprezentowany w obiekcie SMO przez element JMSHeader, który zawiera wszystkie pola znajdujące się zazwyczaj w nagłówku JMS. Mimo że mogą one być modyfikowane podczas mediacji (lub w funkcji ContextService), niektóre pola nagłówka systemu JMS ustawione w obiekcie SMO nie będą propagowane w wychodzącym komunikacie JMS, ponieważ są nadpisywane przez wartości systemowe lub statyczne.

Kluczowymi polami w nagłówku systemu JMS, które mogą być aktualizowane podczas mediacji (lub w funkcji ContextService), są:

- **JMSType** i **JMSCorrelationID** – wartości konkretnych właściwości predefiniowanego nagłówka komunikatu.
- **JMSDeliveryMode** – wartości dla trybu dostarczania (persistent - trwały lub nonpersistent - nietrwały; wartość domyślna to persistent).

- **JMSPriority** – wartość priorytetu (od 0 do 9, wartość domyślna to JMS_Default_Priority).

Właściwości JMS

Właściwości JMS są reprezentowane w obiekcie SMO jako pozycje na liście Properties (właściwości). Właściwości można dodawać, aktualizować lub usuwać podczas mediacji lub przy użyciu funkcji API ContextService.

Właściwości mogą być też ustawiane statycznie w powiązaniu JMS. Właściwości ustawione statycznie nadpisują ustawienia (o tej samej nazwie), które są ustawione dynamicznie.

Właściwości użytkownika propagowane z innych powiązań (na przykład powiązania HTTP) będą wyprowadzane w powiązaniu JMS jako właściwości JMS.

Ustawienia propagowania nagłówka

Propagowanie nagłówka systemu JMS i właściwości z przychodzącego komunikatu JMS do kolejnych komponentów lub z wcześniejszych komponentów do wychodzącego komunikatu JMS może być kontrolowane za pośrednictwem flagi Propaguj nagłówki protokołu w powiązaniu.

Jeśli flaga Propaguj nagłówki protokołu jest ustawiona, informacje w nagłówku mogą przepływać do komunikatu lub do komponentu docelowego, jak opisano poniżej:

- Żądanie eksportu JMS

Nagłówek JMS odebrany w komunikacie będzie propagowany do komponentów docelowych za pośrednictwem usługi kontekstu. Właściwości JMS odebrane w komunikacie będą propagowane do komponentów docelowych za pośrednictwem usługi kontekstu.

- Odpowiedź eksportu JMS

Wszystkie pola nagłówka JMS ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu eksportu JMS. Wszystkie właściwości ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu eksportu JMS.

- Żądanie importu JMS

Wszystkie pola nagłówka JMS ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu importu JMS. Wszystkie właściwości ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu importu JMS.

- Odpowiedź importu JMS

Nagłówek JMS odebrany w komunikacie będzie propagowany do komponentów docelowych za pośrednictwem usługi kontekstu. Właściwości JMS odebrane w komunikacie będą propagowane do komponentów docelowych za pośrednictwem usługi kontekstu.

Schemat korelacji tymczasowego, dynamicznego miejsca docelowego odpowiedzi JMS:

Schemat korelacji tymczasowego, dynamicznego miejsca docelowego odpowiedzi powoduje utworzenie unikalnej dynamicznej kolejki lub dynamicznego tematu dla każdego wysłanego żądania.

Do określenia rodzaju kolejki lub tematu tymczasowego, dynamicznego miejsca docelowego używane jest statyczne miejsce docelowe odpowiedzi określone w imporcie. Jest ono ustawiane w polu **ReplyTo** żądania, a import JMS nasłuchuje odpowiedzi w tym miejscu docelowym. Po odebraniu odpowiedzi jest ona ponownie umieszczana w kolejce statycznego miejsca docelowego odpowiedzi w celu przetworzenia asynchronicznego. Pole **CorrelationID** odpowiedzi nie jest używane i nie trzeba go ustawiać.

Problemy związane z transakcjami

Jeśli używane jest tymczasowe, dynamiczne miejsce docelowe, odpowiedź musi być konsumowana w tym samym wątku co wysłana odpowiedź. Żądanie musi być wysłane poza globalną transakcją i musi być zatwierdzone, zanim zostanie odebrane przez usługę zaplecza i zanim zostanie zwrócona odpowiedź.

Trwałość

Tymczasowe kolejki dynamiczne są jednostkami krótkotrwałymi i nie gwarantują tego samego poziomu trwałości, co statyczna kolejka lub statyczny temat. Tymczasowa kolejka dynamiczna lub tymczasowy temat dynamiczny nie przetrwają restartu serwera (podobnie jak komunikaty). Po ponownym umieszczeniu komunikatu w kolejce do statycznego miejsca docelowego odpowiedzi, komunikat ten zachowuje zdefiniowaną w nim trwałość.

Limit czasu

Import oczekuje odpowiedzi w tymczasowym, dynamicznym miejscu docelowym odpowiedzi przez ustalony czas. Ten okres jest pobierany z kwalifikatora czasu ważności odpowiedzi architektury SCA, jeśli jest on ustawiony. W przeciwnym razie wartością domyślną tego okresu jest 60 sekund. Jeśli czas oczekiwania zostanie przekroczony, import zgłasza wyjątek `ServiceTimeoutRuntimeException`.

Klienci zewnętrzne:

Serwer może wysyłać komunikaty do klientów zewnętrznych i odbierać komunikaty od nich za pośrednictwem powiązań JMS.

Klient zewnętrzny (na przykład portal WWW lub system EIS) może wysyłać komunikat do modułu SCA na serwerze lub może być wywoływany przez komponent znajdujący się na serwerze.

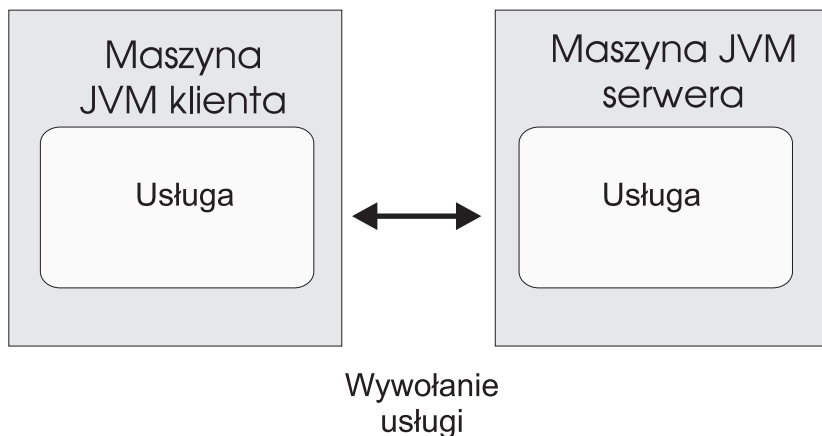
Komponenty eksportu JMS wdrażają procesy nasłuchiwanie komunikatów w celu nasłuchiwanie żądań przychodzących do miejsca docelowego odbierania określonego w powiązaniu eksportu. Miejsce docelowe określone w polu `send` służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołana aplikacja udostępnia odpowiedź. Dzięki temu klient zewnętrzny może wywoływać aplikacje przy użyciu powiązania eksportu.

Importy JMS wchodzi w interakcje z klientami zewnętrznymi, wysyłając komunikaty do kolejek JMS i otrzymując z nich komunikaty.

Praca z klientami zewnętrznymi:

Klient zewnętrzny (znajdujący się poza serwerem) może wymagać interakcji z aplikacją zainstalowaną na serwerze.

Poniżej rozważany jest prosty scenariusz, w którym klient zewnętrzny ma współpracować z pewną aplikacją na serwerze. Na rysunku przedstawiono typowy prosty scenariusz.



Rysunek 34. Scenariusz prostego przypadku użycia: klient zewnętrzny realizuje interakcję z aplikacją serwera

Aplikacja SCA obejmuje element eksportowany z powiązaniem JMS. Dzięki temu aplikacja jest dostępna dla klientów zewnętrznych.

Gdy istnieje klient zewnętrzny wirtualnej maszynie języka Java (JVM), w której nie działa serwer, należy wykonać kilka kroków, aby utworzyć połączenie i zrealizować interakcję z elementem eksportowanym JMS. Klient uzyskuje kontekst początkowy z poprawnymi wartościami i wyszukuje zasoby za pośrednictwem interfejsu JNDI. Następnie klient używa klienta zgodnego ze specyfikacją JMS 1.1 w celu uzyskania dostępu do miejsc docelowych oraz komunikatów o wysłaniu i odbieraniu w miejscach docelowych.

Domyślne nazwy JNDI zasobów tworzonych automatycznie przez środowisko wykonawcze są wymienione w tej sekcji w temacie dotyczącym konfiguracji. Jeśli jednak zasoby zostały utworzone wcześniej, należy używać tych nazw JNDI.

1. Skonfiguruj miejsca docelowe JMS oraz fabrykę połączeń w celu wysłania komunikatu.
2. Upewnij się o poprawności kontekstu JNDI, portu adaptera zasobów SIB oraz portu startowego przesyłania komunikatów.
Serwer używa pewnych portów domyślnych, ale gdy w danym systemie zainstalowanych jest więcej serwerów, podczas instalacji tworzone są porty alternatywne, aby uniknąć konfliktów z innymi instancjami serwerów. Do określania portów używanych przez serwer można używać Konsoli administracyjnej. Należy przejść do opcji **Serwery > Serwery aplikacji > nazwa_używanego_serwera > Konfiguracja** i kliknąć opcję **Porty** w sekcji **Komunikacja**. Następnie można przeprowadzić edycję używanego portu.
3. Klient uzyskuje kontekst początkowy z poprawnymi wartościami i wyszukuje zasoby za pośrednictwem interfejsu JNDI.
4. Za pomocą specyfikacji JMS 1.1 klient uzyskuje dostęp do miejsc docelowych oraz komunikatów o wysłaniu i odbieraniu w miejscach docelowych.

Rozwiązywanie problemów z powiązaniem JMS:

Istnieje możliwość diagnozowania i rozwiązywania problemów związanych z powiązaniem JMS.

Wyjątki implementacji

W odpowiedzi na różne warunki błędów implementacja elementu importowanego lub eksportowanego JMS może zwracać wyjątki dwóch typów:

- Wyjątek biznesowy usługi: ten wyjątek jest zwracany po wystąpieniu uszkodzenia określonego w interfejsie biznesowym usługi (typ portu WSDL).
- Wyjątek środowiska wykonawczego usługi: jest zgłaszany we wszystkich innych przypadkach. Najczęściej wyjątek cause (przyczyna) będzie zawierał pierwotny wyjątek (JMSEception).

Element importowany może na przykład oczekiwać tylko jednego komunikatu odpowiedzi na każdy komunikat żądania. Jeśli nadejdzie więcej niż jedna odpowiedź lub nadejdzie odpowiedź spóźniona (czas utraty ważności odpowiedzi SCA minął), zgłoszony zostanie wyjątek środowiska wykonawczego usługi. Transakcja zostanie wycofana, a komunikat odpowiedzi zostanie usunięty z kolejki lub obsłużony przez menedżer zdarzeń zakończony niepowodzeniem.

Podstawowe warunki niepowodzenia

Podstawowe warunki niepowodzenia powiązań JMS są określane przez semantykę transakcyjną, przez konfigurację dostawcy JMS lub przez odwołanie do aktualnego działania innych komponentów. Istnieją następujące podstawowe warunki niepowodzenia:

- Niepowodzenie nawiązania połączenia z dostawcą JMS lub miejscem docelowym.
Niepowodzenie nawiązania połączenia z dostawcą JMS w celu odbierania komunikatów powoduje, że nie można uruchomić procesu nasłuchiwanie komunikatów. Ten warunek zostaje zarejestrowany w dzienniku produktu WebSphere Application Server. Trwałe komunikaty pozostaną w miejscu docelowym do czasu ich pomyślnego pobrania (lub do utraty ważności).
Niepowodzenie nawiązania połączenia z dostawcą JMS w celu wysyłania komunikatów wychodzących powoduje wycofanie transakcji sterującej wysyłaniem.
- Niepowodzenie analizy komunikatu przychodzącego lub budowy komunikatu wychodzącego.
Niepowodzenie w powiązaniu danych lub procedurze obsługi danych powoduje wycofanie transakcji sterującej wykonywanym działaniem.
- Niepowodzenie wysłania komunikatu wychodzącego.
Niepowodzenie wysłania komunikatu powoduje wycofanie odpowiedniej transakcji.
- Wiele komunikatów odpowiedzi lub nieoczekiwane, spóźnione komunikaty odpowiedzi.
Element importowany oczekuje tylko jednego komunikatu odpowiedzi na każdy komunikat żądania. Ponadto poprawny przedział czasu, w którym można odebrać odpowiedź, jest określany przez kwalifikator utraty ważności odpowiedzi SCA znajdujący się w żądaniu. Po nadejściu odpowiedzi lub upływie czasu utraty ważności rekord korelacji zostaje usunięty. W przypadku nadejścia nieoczekiwanej lub spóźnionej odpowiedzi zostaje zgłoszony wyjątek środowiska wykonawczego usługi.
- Wyjątek środowiska wykonawczego związany z przekroczeniem limitu czasu usługi spowodowany przez spóźnioną odpowiedź, gdy używany jest tymczasowy, dynamiczny schemat korelacji miejsca docelowego odpowiedzi.
Przekroczenie limitu czasu elementu importowanego JMS następuje po upływie czasu określonego przez kwalifikator utraty ważności odpowiedzi SCA (gdy nie jest on ustawiony, domyślnie przyjmuje się czas 60 sekund).

Komunikaty SCA oparte na usługach JMS nie są wyświetlane w menedżerze zdarzeń zakończonych niepowodzeniem

Jeśli komunikaty SCA są wynikiem niepowodzenia interakcji JMS, powinny znaleźć się w menedżerze zdarzeń zakończonych niepowodzeniem. Jeśli takie komunikaty nie pojawiają się w menedżerze zdarzeń zakończonych niepowodzeniem, należy upewnić się, że w bazowym miejscu docelowym SIB miejsca docelowego JMS maksymalną liczbę nieudanych dostaw ustawiono na wartość większą niż **1**. Ustawienie wartości **2** lub większej umożliwia interakcję z menedżerem zdarzeń zakończonych niepowodzeniem w trakcie wywołań SCA dotyczących powiązań JMS.

Obsługa wyjątków:

Sposób skonfigurowania powiązania określa sposób obsługi wyjątków wywołanych przez procedury obsługi danych lub powiązania danych. Na działanie systemu podczas zgłoszenia takiego wyjątku dodatkowy wpływ ma rodzaj przepływu mediacji.

Jeśli procedura obsługi danych lub powiązanie danych zostanie wywołane przez powiązanie użytkownika, może wystąpić wiele różnych problemów. Procedura obsługi danych może na przykład odebrać komunikat z uszkodzonym ładunkiem lub może próbować odczytać komunikat w niepoprawnym formacie.

Sposób obsługi takiego wyjątku przez powiązanie użytkownika jest określany na podstawie tego, w jaki sposób zaimplementowano procedurę obsługi danych lub powiązanie danych. Zalecanym sposobem utworzenia powiązania danych jest zaprojektowanie go tak, aby zgłaszało wyjątek **DataBindingException**.

W przypadku wywołania wyjątku czasu wykonywania, między innymi wyjątku **DataBindingException**, mają miejsce następujące działania:

- Jeśli przepływ mediacji jest skonfigurowany jako transakcyjny, komunikat JMS jest domyślnie zapisywany w menedżerze zdarzeń zakończonych niepowodzeniem w celu ręcznego powtórzenia lub usunięcia.

Uwaga: Użytkownik może zmienić tryb odtwarzania dla powiązania tak, aby komunikat był wycofywany zamiast zapisywania go w menedżerze zdarzeń zakończonych niepowodzeniem.

- Jeśli przepływ mediacji nie jest transakcyjny, wyjątek jest rejestrowany, a komunikat zostaje utracony.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, każdy wyjątek procedury obsługi danych jest opakowany w wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

Ogólne powiązania JMS:

Ogólne powiązanie JMS udostępnia połączenia z dostawcami innych firm zgodnymi ze standardem JMS 1.1. Ogólne powiązanie JMS działa w sposób podobny do powiązań JMS.

Usługa udostępniana za pośrednictwem powiązania JMS umożliwia modułowi SCA (Service Component Architecture) wywoływanie systemów zewnętrznych lub odbieranie komunikatów z tych systemów. System może być zewnętrznym systemem JMS.

Ogólne powiązanie JMS umożliwia integrację z dostawcami JMS, które nie są zgodne z architekturą JCA 1.5, ale obsługują interfejs JMS 1.1 i implementują opcjonalne narzędzie serwera aplikacji JMS. Ogólne powiązanie JMS obsługuje dostawców JMS (w tym Oracle AQ, TIBCO, SonicMQ, WebMethods oraz BEA WebLogic) nieobsługujących specyfikacji JCA 1.5, ale obsługujących narzędzie serwera aplikacji (Application Server Facility) specyfikacji JMS 1.1. Wbudowany dostawca JMS produktu WebSphere (SIBJMS), który jest dostawcą JMS opartym na architekturze JCA 1.5, nie jest obsługiwany przez to powiązanie. W przypadku używania tego dostawcy należy użyć powiązań omówionych w sekcji "Powiązania JMS" na stronie 116.

Tego powiązania ogólnego należy używać w środowisku SCA podczas przeprowadzania integracji z systemem opartym na interfejsie JMS, który nie jest zgodny z architekturą JCA 1.5. Po jego zastosowaniu docelowe aplikacje zewnętrzne mogą otrzymywać i wysyłać komunikaty w celu integracji z komponentem SCA.

Przegląd ogólnych powiązań JMS:

Ogólne powiązania JMS są powiązaniem JMS, które nie są oparte na architekturze JCA i które udostępniają połączenia między środowiskiem SCA (Service Component Architecture) oraz systemami JMS zgodnymi ze standardem 1.1 i implementującymi opcjonalne narzędzie serwera aplikacji JMS.

Ogólne powiązania JMS

Do podstawowych cech ogólnych powiązań importu i eksportu JMS należą następujące:

- Port nasłuchiwania: umożliwia dostawcom JMS, które nie są oparte na architekturze JCA, odbieranie komunikatów i rozsyłanie ich do komponentu bean sterowanego komunikatami (Message Driven Bean - MDB).
- Połączenia: hermetyzują wirtualne połączenie między klientem i aplikacją dostawcy.
- Miejsca docelowe: używane przez klienta do określenia celu tworzonych komunikatów lub źródła konsumowanych komunikatów.
- Dane uwierzytelniania: używane w celu zabezpieczenia dostępu do powiązania.

Ogólne powiązanie importu JMS

Ogólne powiązania importu JMS pozwalają komponentom w module SCA komunikować się z usługami udostępnianymi przez zewnętrznych dostawców JMS, które nie są zgodne z architekturą JCA 1.5.

Częścią odpowiedzialną za połączenie w imporcie JMS jest fabryka połączeń. Fabryka połączeń, czyli obiekt używany przez klienta do tworzenia połączenia z dostawcą, hermetyzuje zbiór parametrów konfiguracyjnych połączenia zdefiniowanych przez administratora. Każda fabryka połączeń jest instancją interfejsu `ConnectionFactory`, `QueueConnectionFactory` lub `TopicConnectionFactory`.

Interakcja z zewnętrznymi systemami JMS obejmuje używanie miejsc docelowych do wysyłania żądań i odbierania odpowiedzi.

Obsługiwane są dwa typy scenariuszy używania dla ogólnego powiązania importu JMS, zależne od typu wywoływanej operacji:

- **Jednokierunkowy:** ogólny import JMS umieszcza komunikat w miejscu docelowym wysyłania skonfigurowanym w powiązaniu importu. Do pola `replyTo` nagłówka JMS nie są wysyłane żadne informacje.
- **Dwukierunkowy (żądanie-odpowiedź):** ogólny import JMS umieszcza komunikat w miejscu docelowym wysyłania, a następnie utrwała odpowiedź otrzymaną z komponentu SCA.

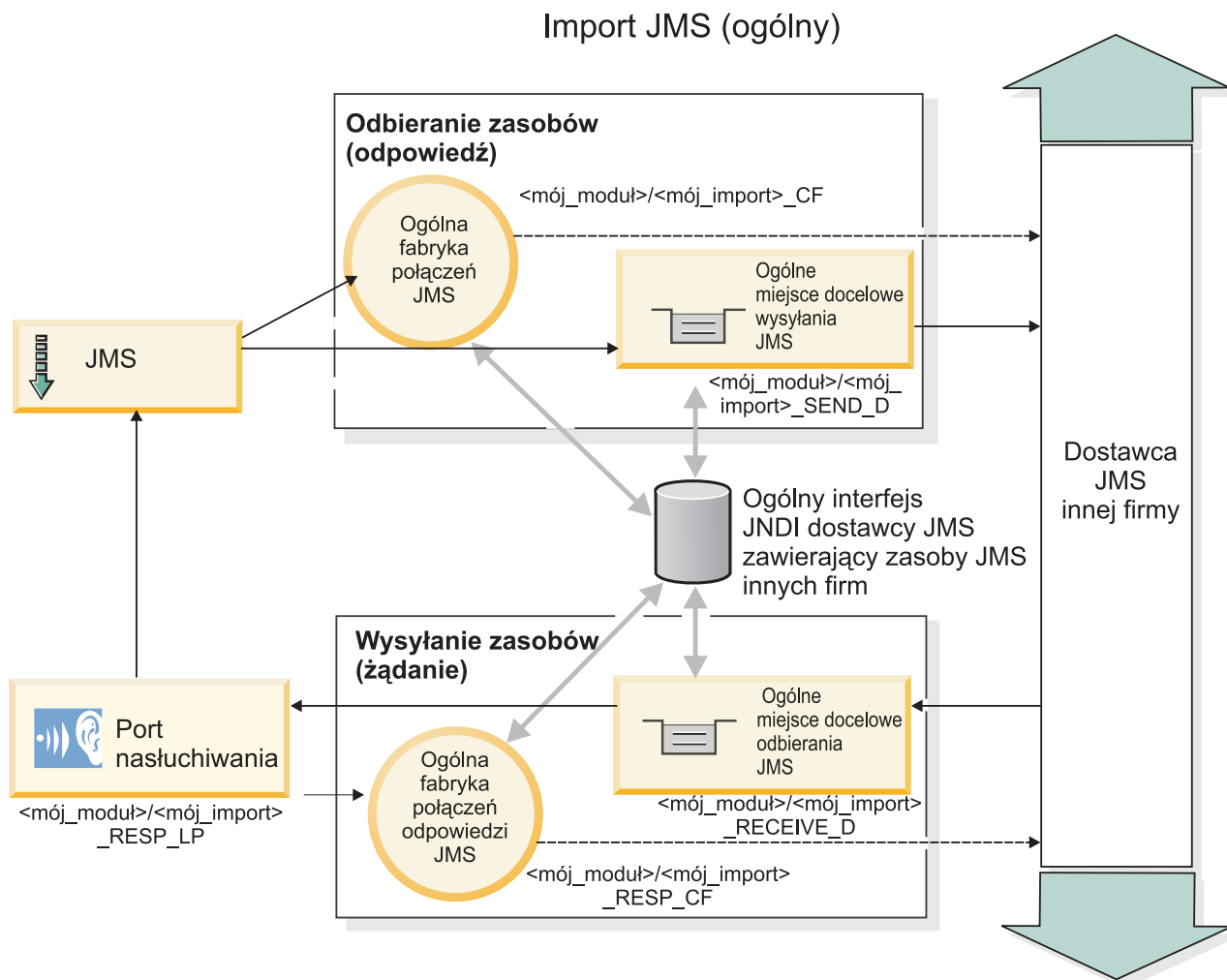
Miejsce docelowe `receive` (odbioru) jest ustawiane we właściwości nagłówka `replyTo` komunikatu wychodzącego. Wdrażany jest komponent bean sterowany komunikatami (Message Driven Bean - MDB), który nasłuchuje w miejscu docelowym odbioru, a po otrzymaniu odpowiedzi ten komponent MDB przekazuje odpowiedź z powrotem do komponentu.

Powiązanie importu może być skonfigurowane (przy użyciu pola **Schemat korelacji odpowiedzi** w produkcie Integration Designer) tak, aby oczekiwać skopiowania identyfikatora korelacji komunikatu odpowiedzi z identyfikatora komunikatu żądania (jest to ustawienie domyślne) lub z identyfikatora korelacji komunikatu żądania.

Istnieje możliwość określenia dynamicznych i statycznych właściwości nagłówka zarówno dla jednokierunkowych, jak i dwukierunkowych scenariuszy używania. Właściwości statyczne można ustawić z poziomu powiązania metody ogólnego importu JMS. Niektóre z tych właściwości mają specjalne znaczenie dla środowiska wykonawczego JMS architektury SCA.

Należy pamiętać, że ogólne powiązanie JMS jest powiązaniem asynchronicznym. Jeśli komponent wywołujący wywołuje ogólny import JMS w sposób synchroniczny (dla operacji dwukierunkowej), komponent wywołujący jest blokowany do chwili zwrócenia odpowiedzi przez usługę JMS.

Na ilustracji Rys. 35 na stronie 126 przedstawiono sposób połączenia importu z usługą zewnętrzną.



Rysunek 35. Zasoby ogólnego powiązania importu JMS

Ogólne powiązania eksportu JMS

Ogólne powiązania eksportu JMS zapewniają modułowi SCA sposób na udostępnianie usług zewnętrznym aplikacjom JMS.

Część odpowiedzialna za połączenie w eksporcie JMS składa się z fabryki połączeń i portu nasłuchiwania.

Ogólny eksport JMS ma miejsca docelowe wysyłania i odbioru.

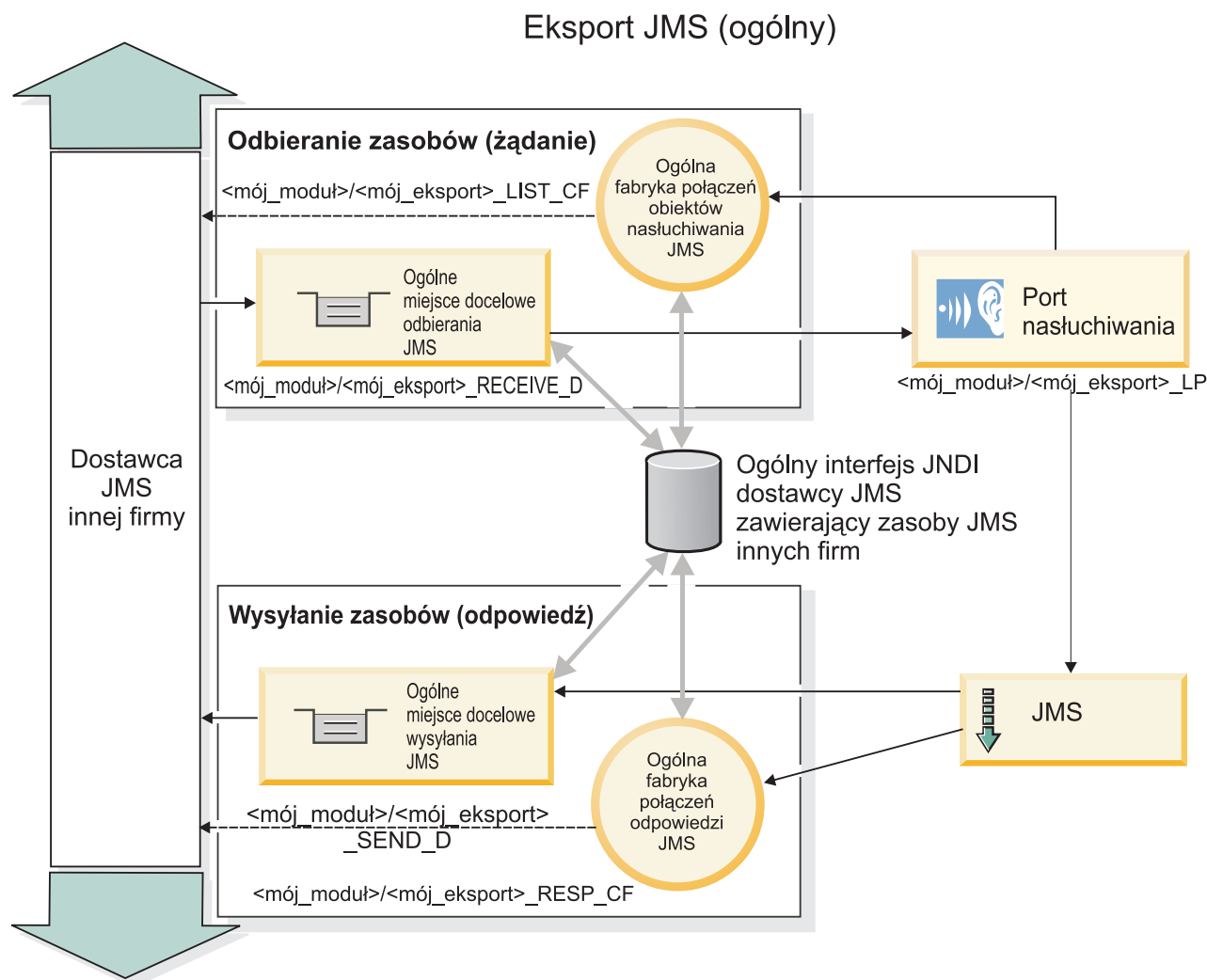
- Miejsce docelowe receive (odbioru) jest miejscem, w którym powinny być umieszczane komunikaty przychodzące do komponentu docelowego.
- Miejsce docelowe send (wysyłania) jest miejscem, do którego wysyłana będzie odpowiedź, chyba że komunikat przychodzący nadpisuje to miejsce przy użyciu właściwości nagłówka replyTo.

Wdrażany jest komponent MDB, który nasłuchuje żądań przychodzących do miejsca docelowego receive (odbioru) określonego w powiązaniu eksportu.

- Miejsce docelowe określone w polu send służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołany komponent udostępni odpowiedź.
- Miejsce docelowe określone w polu replyTo komunikatu przychodzącego nadpisuje miejsce docelowe określone w polu send.

- W przypadku scenariuszy żądania-odpowiedzi powiązanie importu może być skonfigurowane (przy użyciu pola **Schemat korelacji odpowiedzi** w produkcie Integration Designer) tak, aby oczekiwano że odpowiedź skopiuje wartość pola message ID żądania do pola correlation ID komunikatu odpowiedzi (jest to ustawienie domyślne) lub tak, aby oczekiwano że odpowiedź skopiuje wartość pola correlation ID żądania do pola correlation ID komunikatu odpowiedzi.

Na ilustracji Rys. 36 przedstawiono sposób połączenia zewnętrznego requestera z eksportem.



Rysunek 36. Zasoby ogólnego powiązania eksportu JMS

Kluczowe opcje ogólnych powiązań usług JMS:

Opcje ogólnych powiązań elementów importowanych i eksportowanych JMS są spójne z opcjami powiązań osadzonych elementów importowanych JMS i MQ JMS produktu WebSphere. Kluczowe opcje obejmują definicje nagłówek i dostęp do istniejących zasobów środowiska Java EE. Jednak z uwagi na ogólną naturę nie istnieją opcje połączeń specyficzne dla określonego dostawcy JMS, a ponadto możliwości generowania przez te powiązania zasobów podczas wdrażania i instalowania są ograniczone.

Ogólne elementy importowane

Podobnie jak w przypadku aplikacji elementu importowanego MQ JMS ogólna implementacja usługi JMS jest asynchroniczna i obsługuje trzy sposoby wywołania: jednokierunkowe, dwukierunkowe (nazywane także żądanie-odpowiedź) oraz wywołanie zwrotne.

Podczas wdrażania elementu importowanego JMS wdrożony zostaje także sterowany komunikatami komponent bean (MDB) udostępniany przez środowisko wykonawcze. Komponent MDB nasłuchuje odpowiedzi na komunikat żądania. Komponent MDB jest powiązany z miejscem docelowym (nasłuchuje go) wysłanym z żądaniem w polu nagłówka replyTo komunikatu JMS.

Ogólne elementy eksportowane

Powiązania ogólnych elementów eksportowanych JMS różnią się od powiązań elementów eksportowanych systemów EIS sposobem zwracania wyniku. Ogólny element eksportowany JMS jawnie wysyła odpowiedź do miejsca docelowego replyTo określonego w komunikacie przychodzącym. Gdy nie jest ono określone, używane jest miejsce docelowe wysyłania.

Podczas wdrażania ogólnego elementu eksportowanego JMS wdrożony zostaje także sterowany komunikatami komponent bean (różny od komponentu MDB używanego na potrzeby ogólnego elementu importowanego JMS). Nasłuchuje on żądań przychodzących w miejscu docelowym odbioru, a następnie rozsyła żądania do przetworzenia przez środowisko wykonawcze SCA.

Nagłówki specjalne

W ogólnych elementach importowanych i eksportowanych JMS stosowane są specjalne właściwości nagłówka zawierające informacje pozwalające powiązaniu docelowemu na właściwą obsługę komunikatu.

Na przykład właściwość TargetFunctionName jest używana przez domyślny selektor funkcji do identyfikowania nazwy operacji w wywoływanym interfejsie elementu eksportowanego.

Uwaga: Powiązanie elementu importowanego można skonfigurować w taki sposób, aby ustawiało nagłówek TargetFunctionName jako nazwę dla każdej operacji.

Zasoby środowiska Java EE

Podczas wdrażania powiązania usługi JMS w środowisku Java EE utworzona zostaje pewna liczba zasobów środowiska Java EE.

- Port procesu nasłuchiwanie na potrzeby nasłuchiwanie miejsca docelowego odbioru (odpowiedź; tylko wywołania dwukierunkowe) w przypadku elementów importowanych oraz miejsca docelowego odbioru (żądanie) w przypadku elementów eksportowanych
- Fabryka połączeń ogólnych JMS na potrzeby połączeń wychodzących (element importowany) oraz połączeń przychodzących (element eksportowany)
- Ogólne miejsce docelowe JMS na potrzeby miejsc docelowych wysyłania (element importowany) i odbioru (element eksportowany; tylko wywołania dwukierunkowe)
- Fabryka połączeń ogólnych JMS na potrzeby połączeń odpowiedzi (tylko wywołania dwukierunkowe i opcjonalne; w przeciwnym razie w przypadku elementów importowanych używane jest połączenie wychodzące, a w przypadku elementów eksportowanych używane jest połączenie przychodzące)
- Ogólne miejsce docelowe JMS na potrzeby miejsc docelowych odbioru (element importowany) i wysyłania (element eksportowany) - tylko wywołania dwukierunkowe
- Domyślne miejsce docelowe wywołań zwrotnych JMS dostawcy przesyłania komunikatów (tylko wywołania dwukierunkowe)
- Domyślna fabryka połączeń wywołań zwrotnych JMS dostawcy przesyłania komunikatów używana w celu uzyskania dostępu do miejsca docelowego wywołań zwrotnych JMS (tylko wywołania dwukierunkowe)
- Miejsce docelowe kolejki wywołań zwrotnych magistrali SIB używane do zapisywania informacji o komunikacie żądania podczas przetwarzania odpowiedzi (tylko wywołania dwukierunkowe)

Czynność instalacji powoduje utworzenie zasobu fabryki połączeń, trzech miejsc docelowych oraz specyfikacji aktywowania na podstawie informacji zawartych w plikach importu i eksportu.

Ogólne nagłówki JMS:

Ogólne nagłówki JMS są obiektami SDO, które zawierają wszystkie właściwości dla ogólnych właściwości komunikatu JMS. Te właściwości mogą pochodzić z komunikatu przychodzącego lub mogą być właściwościami, które zostaną zastosowane do komunikatu wychodzącego.

Komunikat JMS zawiera dwa typy nagłówków: nagłówek systemowy JMS i wiele właściwości JMS. Dostęp do obu typów nagłówków można uzyskać w module mediacji, w obiekcie SMO lub przy użyciu funkcji API ContextService.

Następujące właściwości są ustawione statycznie w klasie methodBinding:

- JMSType,
- JMSCorrelationID,
- JMSDeliveryMode,
- JMSPriority.

Ogólne powiązanie JMS obsługuje również dynamiczne modyfikowanie nagłówków JMS i właściwości w ten sam sposób, co powiązania JMS i MQ JMS.

Niektórzy ogólni dostawcy JMS nakładają ograniczenia właściwości, które mogą być ustawiane przez aplikację oraz określonych kombinacji tych właściwości. Więcej informacji na ten temat należy uzyskać z dokumentacji produktu innej firmy. Do klasy methodBinding została jednak dodana właściwość ignoreInvalidOutboundJMSProperties, która umożliwia propagowanie dowolnych wyjątków.

Ogólne nagłówki JMS i właściwości komunikatu są używane tylko, jeśli włączony jest podstawowy przełącznik powiązania SCDL architektury SCA. Jeśli ten przełącznik jest włączony, propagowane są informacje o kontekście. Domyślnie ten przełącznik jest włączony. Aby uniemożliwić propagowanie informacji o kontekście, należy zmienić jego wartość na **false**.

Jeśli propagowanie kontekstu jest włączone, informacje w nagłówku mogą przepływać do komunikatu lub do komponentu docelowego. Aby włączyć lub wyłączyć propagowanie kontekstu, należy ustawić wartość **true** lub **false** atrybutu contextPropagationEnabled powiązania importu i eksportu. Na przykład:

```
<esbBinding xsi:type="eis:JMSImportBinding" contextPropagationEnabled="true">
```

Wartością domyślną jest **true**.

Rozwiązywanie problemów z ogólnymi powiązaniem JMS:

Istnieje możliwość diagnozowania i rozwiązywania problemów związanych z ogólnym powiązaniem JMS.

Wyjątki implementacji

W odpowiedzi na różne warunki błędu implementacja ogólnego elementu importowanego lub eksportowanego JMS może zwracać wyjątki dwóch typów:

- Wyjątek biznesowy usługi: ten wyjątek jest zwracany po wystąpieniu uszkodzenia określonego w interfejsie biznesowym usługi (typ portu WSDL).
- Wyjątek środowiska wykonawczego usługi: jest zgłaszany we wszystkich innych przypadkach. Najczęściej wyjątek cause (przyczyna) będzie zawierać pierwotny wyjątek (JMSException).

Rozwiązywanie problemów z utratą ważności ogólnego komunikatu JMS

Komunikat żądania wysłany przez dostawcę JMS może utracić ważność.

Utrata ważności żądania odnosi się do utraty ważności komunikatu żądania dostawcy JMS po upłygnięciu czasu `JMSExpiration` określonego dla tego komunikatu żądania. Tak jak w przypadku innych powiązań JMS ogólne powiązanie JMS obsługuje utratę ważności żądania przez ustawianie czasu utraty ważności w komunikacie wywołania zwrotnego generowanym przez element importowany na taką samą wartość, jaką ma żądanie wychodzące. Powiadomienie o utracie ważności komunikatu wywołania zwrotnego wskazuje, że komunikat żądania utracił ważność i należy powiadomić klienta za pomocą wyjątku biznesowego.

Jeśli jednak miejscem docelowym wywołania zwrotnego jest dostawca innej firmy, ten typ określania utraty ważności żądania nie jest obsługiwany.

Utrata ważności odpowiedzi odnosi się do utraty ważności komunikatu odpowiedzi dostawcy JMS po upłygnięciu czasu `JMSExpiration` określonego dla tego komunikatu odpowiedzi.

Określanie czasu utraty ważności odpowiedzi związanej z ogólnym powiązaniem JMS nie jest obsługiwane, ponieważ dokładna realizacja utraty ważności dostawcy JMS innej firmy nie jest zdefiniowana. Istnieje jednak możliwość sprawdzenia, czy odpowiedź nie utraciła ważności, po jej ewentualnym odebraniu.

W przypadku wychodzących komunikatów żądania wartość `JMSExpiration` jest obliczana w oparciu o czas oczekiwania oraz wartości `requestExpiration` ustawione w nagłówku `asyncHeader`.

Rozwiązywanie problemów z ogólną fabryką połączeń JMS

Jeśli zdefiniowano pewne typy fabryk połączeń w ogólnym dostawcy JMS, podczas próby uruchomienia aplikacji można odebrać komunikat o błędzie. W celu uniknięcia problemu można zmodyfikować zewnętrzną fabrykę połączeń.

Podczas uruchamiania aplikacji może zostać wyświetlony następujący komunikat o błędzie:

```
MDB Listener Port JMSConnectionFactory type does not match JMSDestination type  
(Typ fabryki połączeń JMS portu nasłuchiwania MDB jest niezgodny z typem miejsca docelowego JMS)
```

Ten problem może wystąpić wtedy, gdy definiowane są zewnętrzne fabryki połączeń. W szczególności może zostać zgłoszony wyjątek, gdy użytkownik tworzy fabrykę połączeń tematów JMS 1.0.2 zamiast ujednoczonej fabryki połączeń JMS 1.1 (czyli takiej, w której możliwa jest obsługa komunikacji zarówno w trybie punkt z punktem, jak i w trybie publikowania/subskrybowania).

Aby rozwiązać ten problem, wykonaj następujące kroki:

1. Uzyskaj dostęp do używanego ogólnego dostawcy JMS.
2. Zastąp zdefiniowaną fabrykę połączeń tematów JMS 1.0.2 ujednoczoną fabryką połączeń JMS 1.1.

Po uruchomieniu aplikacji z nowo zdefiniowaną fabryką połączeń JMS 1.1 komunikat o błędzie nie powinien być więcej wyświetlany.

Ogólne komunikaty SCA oparte na usługach JMS nie są wyświetlane w menedżerze zdarzeń zakończonych niepowodzeniem

Jeśli komunikaty SCA są wynikiem niepowodzenia ogólnej interakcji JMS, powinny znaleźć się w menedżerze zdarzeń zakończonych niepowodzeniem. Jeśli takie komunikaty nie pojawiają się w menedżerze zdarzeń zakończonych niepowodzeniem, należy upewnić się, że wartość właściwości określającej maksymalną liczbę ponowionych prób bazowego portu nasłuchiwania jest równa lub większa niż 1. Ustawienie wartości 1 lub większej umożliwia interakcję z menedżerem zdarzeń zakończonych niepowodzeniem w trakcie wywołań SCA związanych z ogólnymi powiązaniem JMS.

Obsługa wyjątków:

Sposób skonfigurowania powiązania określa sposób obsługi wyjątków wywołanych przez procedury obsługi danych lub powiązania danych. Na działanie systemu podczas zgłoszenia takiego wyjątku dodatkowo wpływ ma rodzaj przepływu mediacji.

Jeśli procedura obsługi danych lub powiązanie danych zostanie wywołane przez powiązanie użytkownika, może wystąpić wiele różnych problemów. Procedura obsługi danych może na przykład odebrać komunikat z uszkodzonym ładunkiem lub może próbować odczytać komunikat w niepoprawnym formacie.

Sposób obsługi takiego wyjątku przez powiązanie użytkownika jest określany na podstawie tego, w jaki sposób zaimplementowano procedurę obsługi danych lub powiązanie danych. Zalecanym sposobem utworzenia powiązania danych jest zaprojektowanie go tak, aby zgłaszało wyjątek **DataBindingException**.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, każdy wyjątek procedury obsługi danych jest opakowany w wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

W przypadku wywołania wyjątku czasu wykonywania, między innymi wyjątku **DataBindingException**, mają miejsce następujące działania:

- Jeśli przepływ mediacji jest skonfigurowany jako transakcyjny, komunikat JMS jest domyślnie zapisywany w menedżerze zdarzeń zakończonych niepowodzeniem w celu ręcznego powtórzenia lub usunięcia.

Uwaga: Użytkownik może zmienić tryb odtwarzania dla powiązania tak, aby komunikat był wycofywany zamiast zapisywania go w menedżerze zdarzeń zakończonych niepowodzeniem.

- Jeśli przepływ mediacji nie jest transakcyjny, wyjątek jest rejestrowany, a komunikat zostaje utracony.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, wyjątek procedury obsługi danych jest tworzony wewnątrz wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

Powiązania WebSphere MQ JMS:

Powiązanie WebSphere MQ JMS udostępnia integrację z zewnętrznymi aplikacjami, które używają dostawcy opartego na specyfikacji WebSphere MQ JMS.

Powiązania eksportu i importu WebSphere MQ JMS służą do bezpośredniej integracji z zewnętrznymi systemami JMS lub MQ JMS z poziomu środowiska serwera użytkownika. To eliminuje konieczność używania opcji łącza MQ lub łącza klienta magistrali integracji usług.

Jeśli komponent współdziała z usługą opartą na specyfikacji WebSphere MQ JMS za pośrednictwem importu, powiązanie importu WebSphere MQ wykorzystuje miejsce docelowe, do którego będą wysyłane dane oraz miejsce docelowe, w którym może odebrać odpowiedź. Konwersja danych na komunikat JMS i z komunikatu JMS jest realizowana za pośrednictwem komponentu krańcowego procedury obsługi danych JMS lub powiązania danych JMS.

Jeśli moduł SCA udostępnia usługę klientom WebSphere MQ JMS, powiązanie eksportu WebSphere MQ JMS wykorzystuje miejsce docelowe, w którym może odebrać żądanie oraz z którego może być wysłana odpowiedź. Konwersja danych na komunikat JMS i z komunikatu JMS jest realizowana za pośrednictwem procedury obsługi danych JMS lub powiązania danych JMS.

Selektor funkcji udostępnia odwzorowanie operacji w komponencie docelowym, który ma zostać wywołany.

Przegląd powiązań WebSphere MQ JMS:

Powiązanie WebSphere MQ JMS udostępnia integrację z aplikacjami zewnętrznymi używającymi dostawcy WebSphere MQ JMS.

Czynności administracyjne dotyczące produktu WebSphere MQ

Administrator systemu WebSphere MQ powinien utworzyć bazowy menedżer kolejek WebSphere MQ, który będzie używany przez powiązania WebSphere MQ JMS, zanim zostanie uruchomiona aplikacja zawierająca te powiązania.

Powiązania importu WebSphere MQ JMS

Import WebSphere MQ JMS umożliwia komponentom w module SCA komunikację z usługami udostępnianymi przez dostawców opartych na specyfikacji WebSphere MQ JMS. Użytkownik musi wykorzystywać obsługiwaną wersję produktu WebSphere MQ. Szczegółowe wymagania sprzętowe i programowe zawierają strony działu wsparcia IBM.

Obsługiwane są dwa typy scenariuszy używania dla powiązań importu WebSphere MQ JMS, zależne od typu wywoływanej operacji:

- **Jednokierunkowy:** import WebSphere MQ JMS umieszcza komunikat w miejscu docelowym wysyłania skonfigurowanym w powiązaniu importu. Do pola replyTo nagłówka JMS nie są wysyłane żadne informacje.
- **Dwukierunkowy (żądanie-odpowieź):** import WebSphere MQ JMS umieszcza komunikat w miejscu docelowym wysyłania.

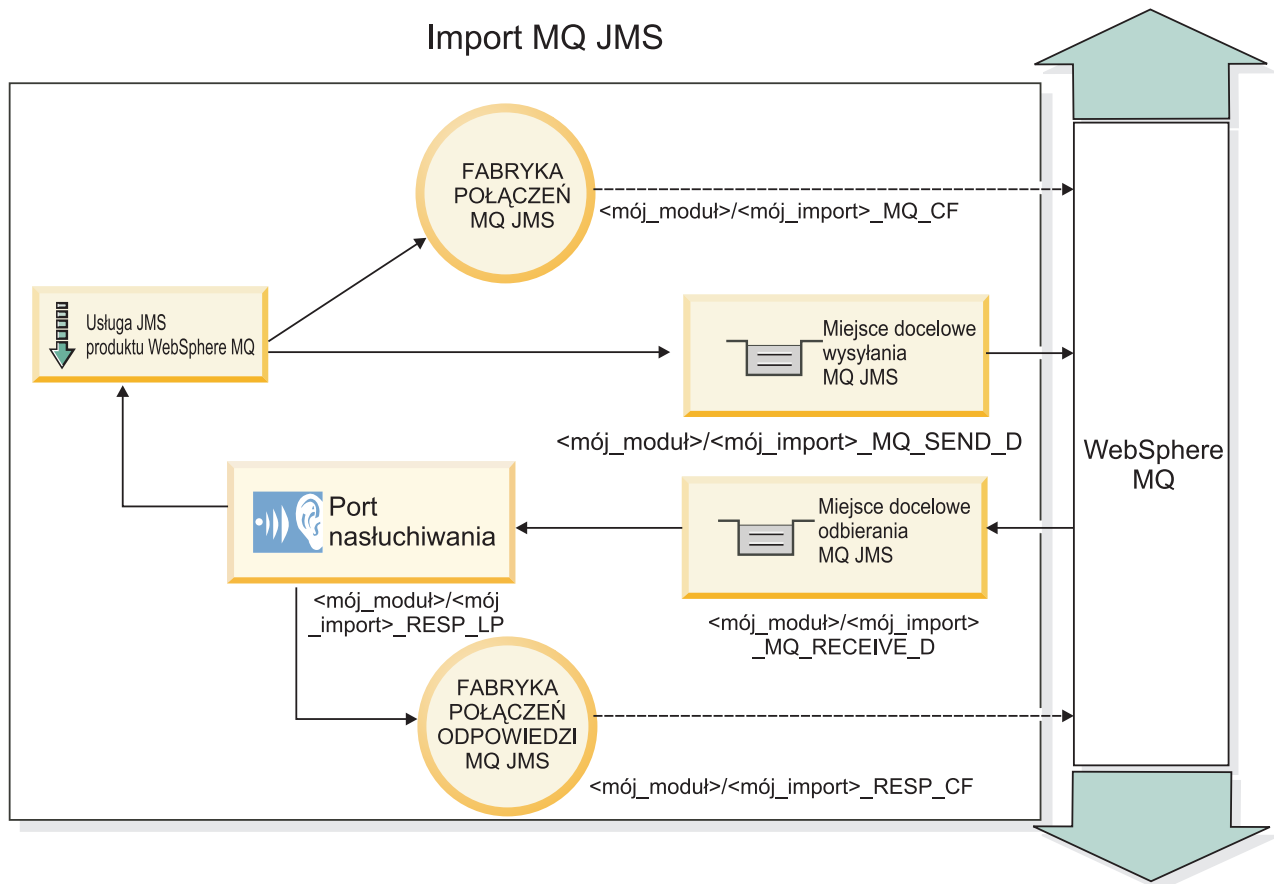
Miejsce docelowe receive (odbioru) jest ustawiane w polu nagłówka replyTo. Wdrażany jest komponent bean sterowany komunikatami (Message Driven Bean - MDB), który nasłuchuje w miejscu docelowym odbioru, a po otrzymaniu odpowiedzi ten komponent MDB przekazuje odpowiedź z powrotem do komponentu.

Powiązanie importu może być skonfigurowane (przy użyciu pola **Schemat korelacji odpowiedzi** w produkcie Integration Designer) tak, aby oczekiwać skopiowania identyfikatora korelacji komunikatu odpowiedzi z identyfikatora komunikatu żądania (jest to ustawienie domyślne) lub z identyfikatora korelacji komunikatu żądania.

Istnieje możliwość określenia dynamicznych i statycznych właściwości nagłówka zarówno dla jednokierunkowych, jak i dwukierunkowych scenariuszy używania. Właściwości statyczne można ustawić z poziomu powiązania metody importu JMS. Niektóre z tych właściwości mają specjalne znaczenie dla środowiska wykonawczego JMS architektury SCA.

Należy pamiętać, że powiązanie WebSphere MQ JMS jest powiązaniem asynchronicznym. Jeśli komponent wywołujący wywołuje import WebSphere MQ JMS w sposób synchroniczny (dla operacji dwukierunkowej), komponent wywołujący jest blokowany do chwili zwrócenia odpowiedzi przez usługę JMS.

Na ilustracji Rys. 37 na stronie 133 przedstawiono sposób połączenia importu z usługą zewnętrzną.



Rysunek 37. Zasoby powiązania importu WebSphere MQ JMS

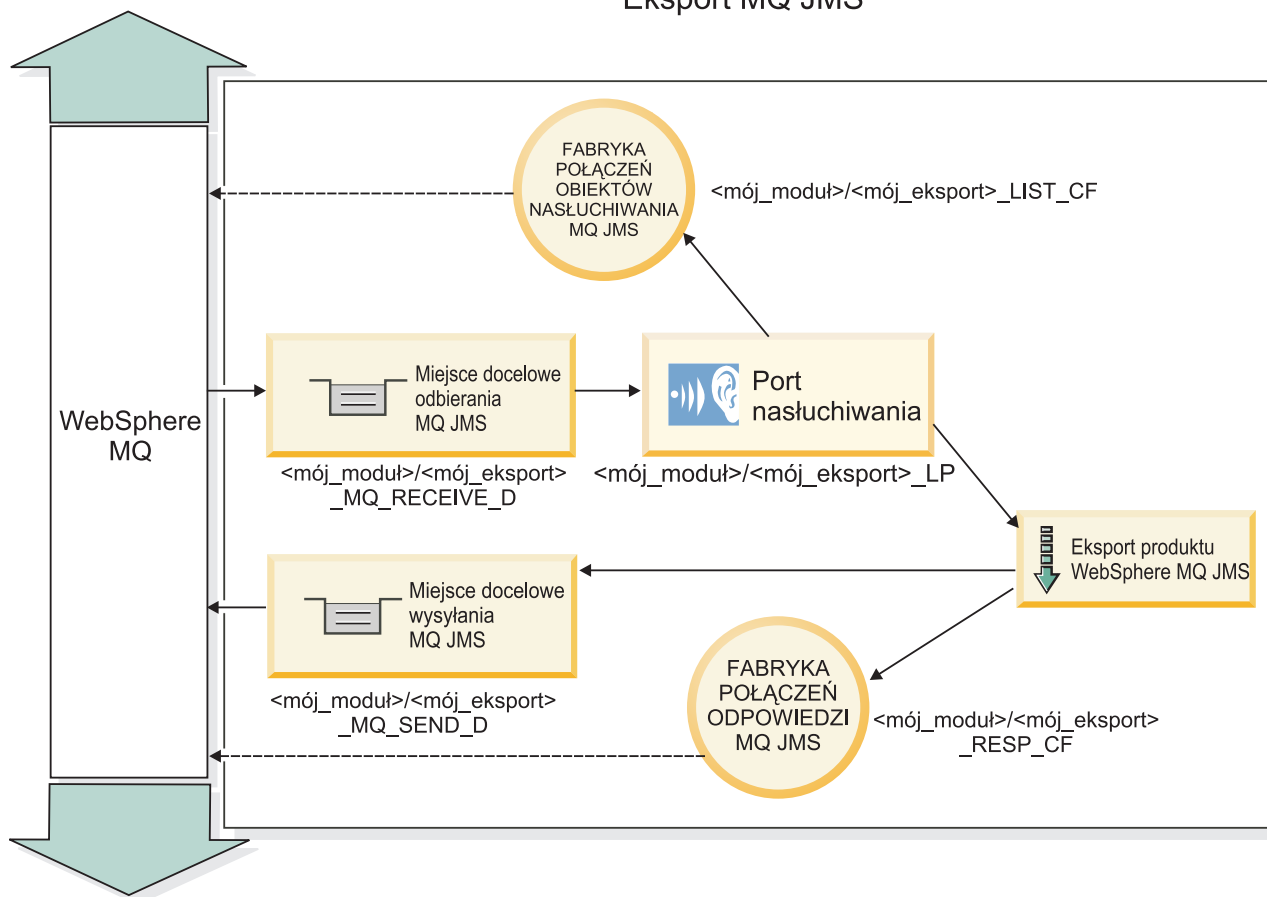
Powiązania eksportu WebSphere MQ JMS

Powiązanie eksportu WebSphere MQ JMS zapewnia modułom SCA sposób udostępniania usług zewnętrznym aplikacjom JMS w dostawcy JMS opartym na produkcie WebSphere MQ.

Wdrażany jest komponent MDB, który nasłuchuje żądań przychodzących do miejsca docelowego receive (odbioru) określonego w powiązaniu eksportu. Miejsce docelowe określone w polu send służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywoływany komponent udostępnia odpowiedź. Miejsce docelowe określone w polu replyTo komunikatu odpowiedzi nadpisuje miejsce docelowe określone w polu send.

Na ilustracji Rys. 38 na stronie 134 przedstawiono sposób połączenia zewnętrznego requestera z eksportem.

Eksport MQ JMS



Rysunek 38. Zasoby powiązania eksportu WebSphere MQ JMS

Uwaga: Ilustracje Rys. 37 na stronie 133 i Rys. 38 przedstawiają sposób połączenia aplikacji z wcześniejszej wersji produktu IBM Business Process Manager z usługą zewnętrzną. W przypadku aplikacji utworzonych dla produktu IBM Business Process Manager 7.0 zamiast portu nasłuchiwanie i fabryki połączeń używana jest specyfikacja aktywowania.

Kluczowe opcje powiązań usług JMS z produktem WebSphere MQ:

Kluczowe opcje powiązań usług JMS z produktem WebSphere MQ obejmują nagłówki, artefakty środowiska Java EE oraz utworzone zasoby środowiska Java EE.

Nagłówki

W nagłówku komunikatu JMS znajduje się pewna liczba predefiniowanych pól zawierających wartości używane zarówno przez klienty, jak i przez dostawców do identyfikowania i kierowania komunikatów. Nagłówki można konfigurować ustalonymi wartościami za pomocą właściwości powiązań lub można je określać dynamicznie w czasie wykonywania.

JMSCorrelationID

Odsyła do komunikatu pokrewnego. Wartością tego pola jest zwykle łańcuch identyfikatora komunikatu, na który dany komunikat jest odpowiedzią.

TargetFunctionName

Ten nagłówek jest używany przez jeden z dostępnych selektorów funkcji do identyfikowania wywoływanej operacji. Ustawienie właściwości nagłówka TargetFunctionName usługi JMS w komunikacie wysłanym do elementu eksportowanego JMS pozwala użyć tego selektora funkcji. Tę właściwość można ustawiać bezpośrednio w aplikacjach klienckich JMS lub podczas nawiązywania połączenia z elementem importowanym z powiązaniem

usługi JMS do takiego elementu eksportowanego. W takim przypadku należy skonfigurować powiązanie elementu importowanego JMS w taki sposób, aby ustawiło nagłówek TargetFunctionName dla każdej operacji w interfejsie na nazwę operacji.

Schematy korelacji

Powiązania usługi JMS z produktem WebSphere MQ udostępniają różne schematy korelacji, które są używane do określania, w jaki sposób należy korelować komunikaty żądania z komunikatami odpowiedzi.

RequestMsgIDToCorrelID

Do pola JMSCorrelationID kopiowana jest wartość pola JMSMessageID. To jest ustawienie domyślne.

RequestCorrelIDToCorrelID

Do pola JMSCorrelationID kopiowana jest wartość pola JMSCorrelationID.

Zasoby środowiska Java EE

Podczas wdrażania elementu importowanego MQ JMS w środowisku Java EE utworzona zostaje pewna liczba zasobów środowiska Java EE.

Parametry

Fabryka połączeń MQ

Używana przez klienty do tworzenia połączenia z dostawcą MQ JMS.

Fabryka połączeń odpowiedzi

Używana przez środowisko wykonawcze MQ JMS architektury SCA, gdy miejsce docelowe wysyłania znajduje się w innym menedżerze kolejek niż miejsce docelowe odbioru.

Specyfikacja aktywowania

Specyfikacja aktywowania MQ JMS jest powiązana z jednym lub kilkoma komponentami bean sterowanymi komunikatami i udostępnia konfigurację pozwalającą im odbierać komunikaty.

Miejsca docelowe

- Miejsce docelowe wysyłania:
 - Elementy importowane: miejsce wysyłania żądania lub komunikatu wychodzącego.
 - Elementy eksportowane: miejsce, do którego jest wysyłany komunikat odpowiedzi, jeśli nie jest zastąpione za pomocą pola nagłówka JMSReplyTo w komunikacie przychodzącym.
- Miejsce docelowe odbioru:
 - Elementy importowane: miejsce, w którym należy umieścić odpowiedź lub komunikat przychodzący.
 - Elementy eksportowane: miejsce, w którym należy umieścić komunikat przychodzący lub żądanie.

Nagłówki JMS:

Komunikat JMS zawiera dwa typy nagłówków: nagłówek systemowy JMS i wiele właściwości JMS. Dostęp do obu typów nagłówków można uzyskać w module mediacji, w obiekcie SMO lub przy użyciu funkcji API ContextService.

Nagłówek systemu JMS

Nagłówek systemu JMS jest reprezentowany w obiekcie SMO przez element JMSHeader, który zawiera wszystkie pola znajdujące się zazwyczaj w nagłówku JMS. Mimo że mogą one być modyfikowane podczas mediacji (lub w funkcji ContextService), niektóre pola nagłówka systemu JMS ustawione w obiekcie SMO nie będą propagowane w wychodzącym komunikacie JMS, ponieważ są nadpisywane przez wartości systemowe lub statyczne.

Kluczowymi polami w nagłówku systemu JMS, które mogą być aktualizowane podczas mediacji (lub w funkcji ContextService), są:

- **JMSType** i **JMSCorrelationID** – wartości konkretnych właściwości predefiniowanego nagłówka komunikatu.

- **JMSDeliveryMode** – wartości dla trybu dostarczania (persistent - trwałe lub nonpersistent - nietrwałe; wartość domyślna to persistent).
- **JMSPriority** – wartość priorytetu (od 0 do 9, wartość domyślna to JMS_Default_Priority).

Właściwości JMS

Właściwości JMS są reprezentowane w obiekcie SMO jako pozycje na liście Properties (właściwości). Właściwości można dodawać, aktualizować lub usuwać podczas mediacji lub przy użyciu funkcji API ContextService.

Właściwości mogą być też ustawiane statycznie w powiązaniu JMS. Właściwości ustawione statycznie nadpisują ustawienia (o tej samej nazwie), które są ustawione dynamicznie.

Właściwości użytkownika propagowane z innych powiązań (na przykład powiązania HTTP) będą wyprowadzane w powiązaniu JMS jako właściwości JMS.

Ustawienia propagowania nagłówka

Propagowanie nagłówka systemu JMS i właściwości z przychodzącego komunikatu JMS do kolejnych komponentów lub z wcześniejszych komponentów do wychodzącego komunikatu JMS może być kontrolowane za pośrednictwem flagi Propaguj nagłówki protokołu w powiązaniu.

Jeśli flaga Propaguj nagłówki protokołu jest ustawiona, informacje w nagłówku mogą przepływać do komunikatu lub do komponentu docelowego, jak opisano poniżej:

- **Żądanie eksportu JMS**

Nagłówek JMS odebrany w komunikacie będzie propagowany do komponentów docelowych za pośrednictwem usługi kontekstu. Właściwości JMS odebrane w komunikacie będą propagowane do komponentów docelowych za pośrednictwem usługi kontekstu.

- **Odpowiedź eksportu JMS**

Wszystkie pola nagłówka JMS ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu eksportu JMS. Wszystkie właściwości ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu eksportu JMS.

- **Żądanie importu JMS**

Wszystkie pola nagłówka JMS ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu importu JMS. Wszystkie właściwości ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu importu JMS.

- **Odpowiedź importu JMS**

Nagłówek JMS odebrany w komunikacie będzie propagowany do komponentów docelowych za pośrednictwem usługi kontekstu. Właściwości JMS odebrane w komunikacie będą propagowane do komponentów docelowych za pośrednictwem usługi kontekstu.

Klienci zewnętrzne:

Serwer może wysyłać komunikaty do klientów zewnętrznych i odbierać komunikaty od nich za pośrednictwem powiązań WebSphere MQ JMS.

Klient zewnętrzny (na przykład portal WWW lub system EIS) może wysyłać komunikat do komponentu SCA w aplikacji za pośrednictwem eksportu lub może być wywoływany przez komponent SCA w aplikacji za pośrednictwem importu.

Powiązanie eksportu WebSphere MQ JMS wdraża komponenty bean sterowane komunikatami (Message Driven Bean - MDB) w celu nasłuchiwania żądań przychodzących do miejsca docelowego odbioru określonego w powiązaniu

eksportu. Miejsce docelowe określone w polu send służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołana aplikacja udostępnia odpowiedź. Dzięki temu klient zewnętrzny może wywoływać aplikacje za pośrednictwem powiązania eksportu.

Importy WebSphere MQ JMS są wiązane z klientami zewnętrznymi i mogą dostarczać do nich komunikaty. Komunikat może wymagać odpowiedzi od klienta zewnętrznego, ale może też nie wymagać tej odpowiedzi.

Więcej informacji na temat interakcji z klientami zewnętrznymi przy użyciu produktu WebSphere MQ można znaleźć w Centrum informacyjnym produktu WebSphere MQ.

Rozwiązywanie problemów z powiązaniem JMS produktu WebSphere MQ:

Istnieje możliwość diagnozowania i rozwiązywania problemów dotyczących powiązań JMS produktu WebSphere MQ.

Wyjątki implementacji

W odpowiedzi na różne warunki błędu implementacja elementu importowanego lub eksportowanego MQ JMS może zwracać wyjątki dwóch typów:

- Wyjątek biznesowy usługi: ten wyjątek jest zwracany po wystąpieniu uszkodzenia określonego w interfejsie biznesowym usługi (typ portu WSDL).
- Wyjątek środowiska wykonawczego usługi: jest zgłaszany we wszystkich innych przypadkach. Najczęściej wyjątek cause (przyczyna) będzie zawierać pierwotny wyjątek (JMSEException).

Element importowany może na przykład oczekiwać tylko jednego komunikatu odpowiedzi na każdy komunikat żądania. Jeśli nadejdzie więcej niż jedna odpowiedź lub nadejdzie odpowiedź spóźniona (czas utraty ważności odpowiedzi SCA minął), zgłoszony zostanie wyjątek środowiska wykonawczego usługi. Transakcja zostanie wycofana, a komunikat odpowiedzi zostanie usunięty z kolejki lub obsłużony przez menedżer zdarzeń zakończonych niepowodzeniem.

Komunikaty SCA oparte na usługach JMS produktu WebSphere MQ nie są wyświetlane w menedżerze zdarzeń zakończonych niepowodzeniem

Jeśli komunikaty SCA są wynikiem niepowodzenia interakcji usług JMS produktu WebSphere MQ, powinny znaleźć się w menedżerze zdarzeń zakończonych niepowodzeniem. Jeśli takie komunikaty nie pojawiają się w menedżerze zdarzeń zakończonych niepowodzeniem, należy upewnić się, że wartość właściwości określającej maksymalną liczbę ponowionych prób bazowego portu nasłuchiwania jest równa lub większa niż 1. Ustawienie wartości 1 lub większej umożliwia interakcję z menedżerem zdarzeń zakończonych niepowodzeniem w trakcie wywołań SCA dotyczących powiązań MQ JMS.

Scenariusze niepoprawnego użycia: porównanie z powiązaniem produktu WebSphere MQ

Powiązanie JMS produktu WebSphere MQ zostało zaprojektowane z myślą o współpracy z aplikacjami JMS wdrożonymi z produktem WebSphere MQ, który ujawnia komunikaty zgodnie z modelem komunikatów JMS. Elementy importowane i eksportowane produktu WebSphere MQ zostały jednak zaprojektowane głównie na potrzeby współpracy z rodzimymi aplikacjami produktu WebSphere MQ i ujawniają mediacjom pełną treść komunikatu produktu WebSphere MQ.

W następujących scenariuszach należy stosować powiązania JMS produktu WebSphere MQ zamiast powiązań produktu WebSphere MQ:

- Wywoływanie komponentu bean sterowanego komunikatami JMS (MDB) z modułu SCA, gdy komponent MDB jest wdrożony w dostawcy JMS produktu WebSphere MQ. Należy użyć elementu importowanego JMS produktu WebSphere MQ.
- Umożliwienie wywołania modułu SCA z serwletu komponentów Java EE lub z komponentu EJB za pomocą usług JMS. Należy użyć elementu eksportowanego JMS produktu WebSphere MQ.

- Mediowanie treści komunikatu JMS MapMessage w trakcie przejścia przez produkt WebSphere MQ. Należy użyć elementu eksportowanego i elementu importowanego JMS produktu WebSphere MQ w połączeniu z odpowiednią procedurą obsługi danych lub odpowiednim powiązaniem danych.

Są takie sytuacje, w których powiązanie produktu WebSphere MQ i powiązanie JMS produktu WebSphere MQ JMS powinny współpracować. W szczególności podczas łączenia aplikacji Java EE i innych niż Java EE w produkcie WebSphere MQ należy używać elementu eksportowanego produktu WebSphere MQ oraz elementu importowanego JMS produktu WebSphere MQ (lub odwrotnie) w połączeniu z odpowiednimi powiązaniem danych lub modułami mediacji (bądź dwoma tymi rodzajami obiektów).

Obsługa wyjątków:

Sposób skonfigurowania powiązania określa sposób obsługi wyjątków wywołanych przez procedury obsługi danych lub powiązania danych. Na działanie systemu podczas zgłoszenia takiego wyjątku dodatkowy wpływ ma rodzaj przepływu mediacji.

Jeśli procedura obsługi danych lub powiązanie danych zostanie wywołane przez powiązanie użytkownika, może wystąpić wiele różnych problemów. Procedura obsługi danych może na przykład odebrać komunikat z uszkodzonym ładunkiem lub może próbować odczytać komunikat w niepoprawnym formacie.

Sposób obsługi takiego wyjątku przez powiązanie użytkownika jest określany na podstawie tego, w jaki sposób zaimplementowano procedurę obsługi danych lub powiązanie danych. Zalecanym sposobem utworzenia powiązania danych jest zaprojektowanie go tak, aby zgłaszało wyjątek **DataBindingException**.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, każdy wyjątek procedury obsługi danych jest opakowany w wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

W przypadku wywołania wyjątku czasu wykonywania, między innymi wyjątku **DataBindingException**, mają miejsce następujące działania:

- Jeśli przepływ mediacji jest skonfigurowany jako transakcyjny, komunikat JMS jest domyślnie zapisywany w menedżerze zdarzeń zakończonych niepowodzeniem w celu ręcznego powtórzenia lub usunięcia.

Uwaga: Użytkownik może zmienić tryb odtwarzania dla powiązania tak, aby komunikat był wycofywany zamiast zapisywania go w menedżerze zdarzeń zakończonych niepowodzeniem.

- Jeśli przepływ mediacji nie jest transakcyjny, wyjątek jest rejestrowany, a komunikat zostaje utracony.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, wyjątek procedury obsługi danych jest tworzony wewnątrz wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

Powiązania produktu WebSphere MQ:

Powiązanie WebSphere MQ udostępnia połączenia architektury SCA (Service Component Architecture) z aplikacjami WebSphere MQ.

Powiązania eksportu i importu WebSphere MQ służą do bezpośredniej integracji z systemem opartym na produkcie WebSphere MQ z poziomu środowiska serwera użytkownika. To eliminuje konieczność używania opcji łącza MQ lub łącza klienta magistrali integracji usług.

Jeśli komponent współdziała z usługą WebSphere MQ za pośrednictwem importu, powiązanie importu WebSphere MQ używa kolejki, do której będą wysyłane dane oraz kolejki, w której może odebrać odpowiedź.

Jeśli moduł SCA udostępnia usługę klientom WebSphere MQ, powiązanie eksportu WebSphere MQ korzysta z kolejki, w której może odebrać żądanie oraz z której może być wysłana odpowiedź. Selektor funkcji udostępnia odwzorowanie operacji w komponencie docelowym, który ma zostać wywołany.

Konwersja danych ładunku z komunikatu MQ i na komunikat MQ jest realizowana za pośrednictwem procedury obsługi danych treści MQ lub powiązania danych treści MQ. Konwersja danych nagłówka na komunikat MQ i z komunikatu MQ jest realizowana za pośrednictwem powiązanie danych nagłówka MQ.

Informacje na temat obsługiwanych wersji produktu WebSphere MQ można znaleźć na stronie WWW zawierającej szczegółowe wymagania systemowe.

Przegląd powiązań WebSphere MQ:

Powiązanie WebSphere MQ udostępnia integrację z rodzimymi aplikacjami opartymi na produkcie MQ.

Czynności administracyjne dotyczące produktu WebSphere MQ

Administrator systemu WebSphere MQ powinien utworzyć bazowy menedżer kolejek WebSphere MQ, który będzie używany przez powiązania WebSphere MQ, zanim zostanie uruchomiona aplikacja zawierająca te powiązania.

Czynności administracyjne dotyczące produktu WebSphere

Należy ustawić właściwość **Rodzima ścieżka do biblioteki** adaptera zasobu MQ w produkcie WebSphere na wartość odpowiadającą wersji produktu WebSphere MQ obsługiwanej przez serwer, a następnie zrestartować serwer. Gwarantuje to, że używane będą biblioteki z obsługiwanej wersji produktu WebSphere MQ. Szczegółowe wymagania sprzętowe i programowe zawierają strony działu wsparcia IBM.

Powiązania importu WebSphere MQ

Powiązanie importu WebSphere MQ umożliwia komponentom w module SCA komunikację z usługami udostępnianymi przez zewnętrzne aplikacje oparte na produkcie WebSphere MQ. Użytkownik musi wykorzystywać obsługiwaną wersję produktu WebSphere MQ. Szczegółowe wymagania sprzętowe i programowe zawierają strony działu wsparcia IBM.

Interakcja z zewnętrznymi systemami WebSphere MQ obejmuje używanie kolejek do wysyłania żądań i odbierania odpowiedzi.

Obsługiwane są dwa typy scenariuszy używania dla powiązań importu WebSphere MQ, zależne od typu wywoływanej operacji:

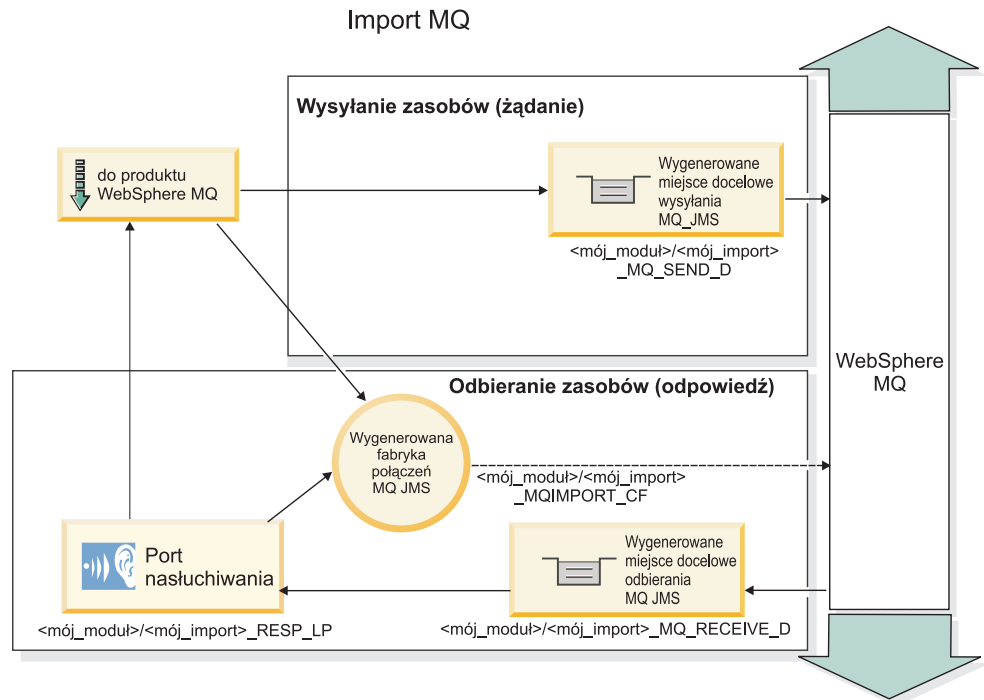
- **Jednokierunkowy:** import WebSphere MQ umieszcza komunikat w kolejce skonfigurowanej w polu **Kolejka miejsca docelowego wysyłania** w powiązaniu importu. Do pola replyTo nagłówka MQMD nie są wysyłane żadne informacje.
- **Dwukierunkowy (żądanie-odpowiedź):** import WebSphere MQ umieszcza komunikat w kolejce skonfigurowanej w polu **Kolejka miejsca docelowego wysyłania**.

Kolejka receive (odbioru) jest ustawiana w polu replyTo nagłówka MQMD. Wdrażany jest komponent bean sterowany komunikatami (Message Driven Bean - MDB), który nasłuchuje w kolejce odbiorczej, a po otrzymaniu odpowiedzi ten komponent MDB przekazuje odpowiedź z powrotem do komponentu.

Powiązanie importu może być skonfigurowane (przy użyciu pola **Schemat korelacji odpowiedzi**) tak, aby oczekiwać skopiowania identyfikatora korelacji komunikatu odpowiedzi z identyfikatora komunikatu żądania (jest to ustawienie domyślne) lub z identyfikatora korelacji komunikatu żądania.

Należy pamiętać, że powiązanie WebSphere MQ jest powiązaniem asynchronicznym. Jeśli komponent wywołujący wywoła import WebSphere MQ w sposób synchroniczny (dla operacji dwukierunkowej), komponent wywołujący jest blokowany do chwili zwrócenia odpowiedzi przez usługę WebSphere MQ.

Na ilustracji Rys. 39 na stronie 140 przedstawiono sposób połączenia importu z usługą zewnętrzną.



Rysunek 39. Zasoby powiązania importu WebSphere MQ

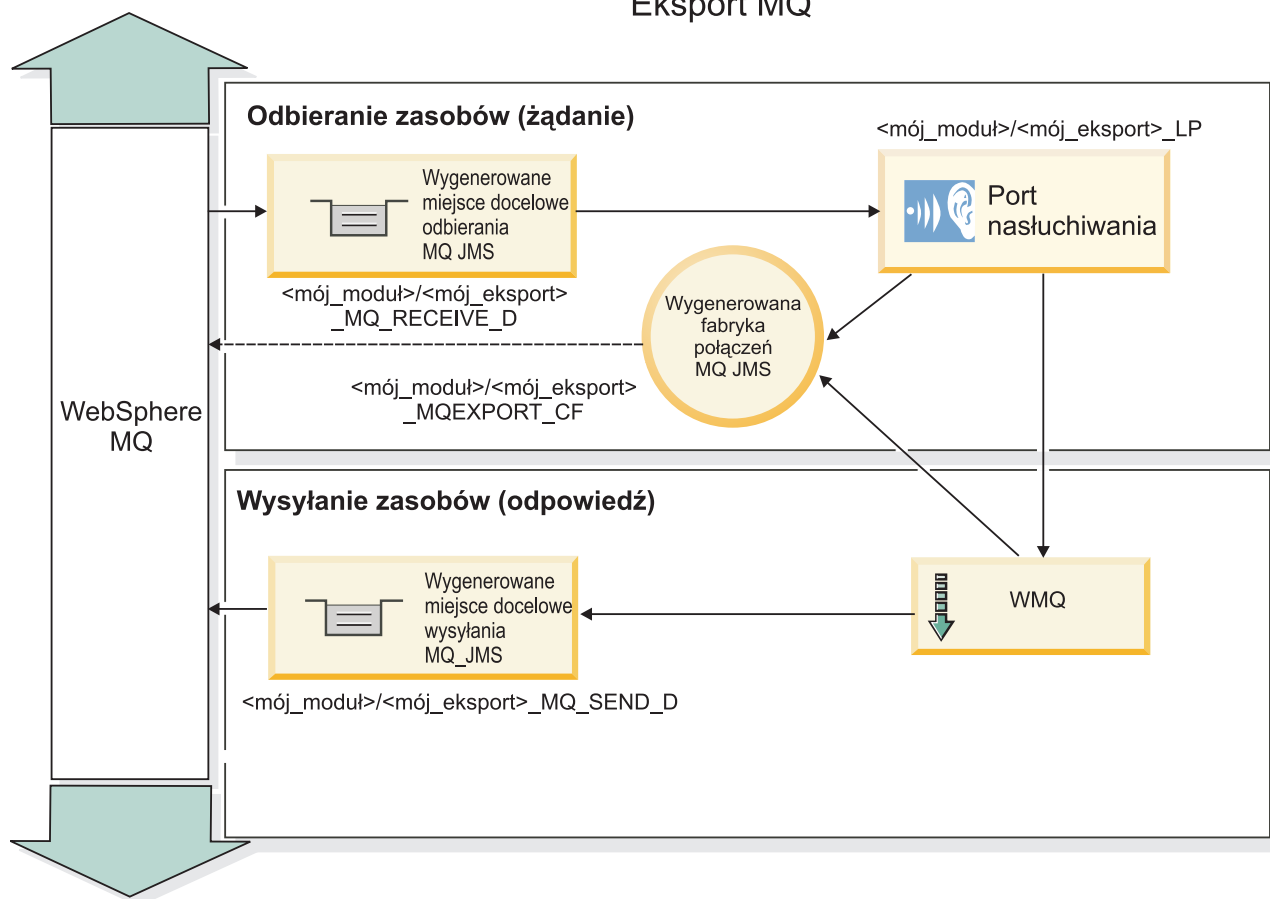
Powiązania eksportu WebSphere MQ

Powiązanie eksportu WebSphere MQ zapewnia modułom SCA sposób udostępniania usług zewnętrznym aplikacjom opartym na produkcie WebSphere MQ.

Wdrażany jest komponent MDB, który nasłuchuje żądań przychodzących do kolejki **Kolejka miejsca docelowego odbioru** określonej w powiązaniu eksportu. Kolejka określona w polu **Kolejka miejsca docelowego wysyłania** służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołany komponent udostępnia odpowiedź. Kolejka określona w polu replyTo komunikatu odpowiedzi nadpisuje kolejkę określoną w polu **Kolejka miejsca docelowego wysyłania**.

Na ilustracji Rys. 40 na stronie 141 przedstawiono sposób połączenia zewnętrznego requestera z eksportem.

Eksport MQ



Rysunek 40. Zasoby powiązania eksportu WebSphere MQ

Uwaga: Ilustracje Rys. 39 na stronie 140 i Rys. 40 przedstawiają sposób połączenia aplikacji z wcześniejszej wersji produktu IBM Business Process Manager z usługą zewnętrzną. W przypadku aplikacji utworzonych dla produktu IBM Business Process Manager w wersji 7.x lub nowszej zamiast portu procesu nasłuchiwania i fabryki połączeń używana jest specyfikacja aktywowania.

Kluczowe opcje powiązania z produktem WebSphere MQ:

Kluczowe opcje powiązania z produktem WebSphere MQ obejmują nagłówki, artefakty środowiska Java EE oraz utworzone zasoby środowiska Java EE.

Schematy korelacji

Aplikacja typu żądanie/odpowiedź produktu WebSphere MQ może używać jednej z wielu technik korelowania komunikatów odpowiedzi z zadaniami korzystających z pól struktury MQMD MessageID i CorrelID. W zdecydowanej większości przypadków requester pozwala menedżerowi kolejek wybrać pole MessageID i oczekuje, że aplikacja odpowiadająca skopiuje jego wartość do pola CorrelID w odpowiedzi. Najczęściej requester i aplikacja odpowiadająca uzyskują niejawnie informacje o aktualnie używanej technice korelacji. Czasami aplikacja odpowiadająca uwzględni różne opcje znajdujące się w polu Report zadaniami opisujące sposób obsługi tych pól.

Powiązania elementów eksportowanych związanych z komunikatami produktu WebSphere MQ można skonfigurować z następującymi opcjami:

Opcje pola MsgId odpowiedzi:

Nowa wartość MsgID

Pozwala menedżerowi kolejek wybrać unikalną wartość pola MsgId w odpowiedzi (opcja domyślna).

Kopiowanie wartości MsgID z żądania

Powoduje skopiowanie wartości pola MsgId z pola MsgId w żądaniu.

Kopiowanie z komunikatu SCA

Powoduje ustawienie wartości pola MsgId na przenoszona w nagłówkach komunikatów odpowiedzi SCA produktu WebSphere MQ lub pozwala menedżerowi kolejek zdefiniować nową wartość Id, gdy ta wartość nie istnieje.

Jak opcje pola Report

Powoduje zbadanie pola Report struktury MQMD w żądaniu w celu wykrycia sposobu obsługi pola MsgId. Obsługiwane są opcje MQRO_NEW_MSG_ID i MQRO_PASS_MSG_ID. Ich działanie jest podobne, jak w przypadku opcji Nowa wartość MsgId i Kopiowanie wartości MsgID z żądania.

Opcje pola CorrelId odpowiedzi:

Kopiowanie wartości MsgID z żądania

Powoduje skopiowanie wartości pola CorrelId z pola MsgId w żądaniu (opcja domyślna).

Kopiowanie wartości CorrelID z żądania

Powoduje skopiowanie wartości pola CorrelId z pola CorrelId w żądaniu.

Kopiowanie z komunikatu SCA

Powoduje ustawienie wartości pola CorrelId na przenoszona w nagłówkach komunikatów odpowiedzi SCA produktu WebSphere MQ lub pozostawienie pustego pola, gdy ta wartość nie istnieje.

Jak opcje pola Report

Powoduje zbadanie pola Report struktury MQMD w żądaniu w celu wykrycia sposobu obsługi pola CorrelId. Obsługiwane są opcje MQRO_COPY_MSG_ID_TO_CORREL_ID i MQRO_PASS_CORREL_ID. Ich działanie jest podobne, jak w przypadku opcji Kopiowanie wartości MsgID z żądania i Kopiowanie wartości CorrelID z żądania.

Powiązania elementów importowanych związanych z komunikatami produktu WebSphere MQ można skonfigurować z następującymi opcjami:

Opcje pola MsgId żądania:

Nowa wartość MsgID

Pozwala menedżerowi kolejek wybrać unikalną wartość pola MsgId w żądaniu (opcja domyślna).

Kopiowanie z komunikatu SCA

Powoduje ustawienie wartości pola MsgId na przenoszona w nagłówkach komunikatów żądania SCA produktu WebSphere MQ lub pozwala menedżerowi kolejek zdefiniować nową wartość Id, gdy ta wartość nie istnieje.

Opcje korelacji odpowiedzi:

Odpowiedź ma wartość CorrelID skopiowaną z pola MsgId

Oczekuje się, że komunikat odpowiedzi będzie mieć ustawioną wartość pola CorrelId na podstawie pola MsgId żądania (opcja domyślna).

Odpowiedź ma wartość MsgID skopiowaną z pola MsgId

Oczekuje się, że komunikat odpowiedzi będzie mieć ustawioną wartość pola MsgId na podstawie pola MsgId żądania.

Odpowiedź ma wartość CorrelID skopiowaną z pola CorrelId

Oczekuje się, że komunikat odpowiedzi będzie mieć ustawioną wartość pola CorrelId na podstawie pola CorrelId żądania.

Zasoby środowiska Java EE

Podczas wdrażania powiązania z produktem WebSphere MQ w środowisku Java EE utworzona zostaje pewna liczba zasobów środowiska Java EE.

Parametry

Fabryka połączeń MQ

Używana przez klienty do tworzenia połączenia z dostawcą WebSphere MQ.

Fabryka połączeń odpowiedzi

Używana przez środowisko wykonawcze MQ architektury SCA, gdy miejsce docelowe wysyłania znajduje się w innym menedżerze kolejek niż miejsce docelowe odbioru.

Specyfikacja aktywowania

Specyfikacja aktywowania MQ JMS jest powiązana z jednym lub kilkoma komponentami bean sterowanymi komunikatami i udostępnia konfigurację pozwalającą im odbierać komunikaty.

Miejsca docelowe

- Miejsce docelowe wysyłania: miejsce, do którego jest wysyłane żądanie lub komunikat wychodzący (element importowany); miejsce, do którego jest wysyłany komunikat odpowiedzi (element eksportowany), jeśli nie jest zastąpione za pomocą pola nagłówka ReplyTo struktury MQMD w komunikacie przychodzącym.
- Miejsce docelowe odbioru: miejsce, w którym należy umieścić odpowiedź/żądanie lub komunikat przychodzący.

Nagłówki WebSphere MQ:

Nagłówki WebSphere MQ uwzględniają określone konwencje podczas konwersji na komunikaty SCA (Service Component Architecture).

Komunikaty WebSphere MQ składają się z nagłówka systemu (struktury MQMD), opcjonalnych innych nagłówków MQ (systemu lub niestandardowych) oraz treści komunikatu. Jeśli w komunikacie istnieje wiele nagłówków komunikatu, kolejność nagłówków ma znaczenie.

Każdy nagłówek zawiera informacje opisujące strukturę kolejnego nagłówka. Struktura MQMD opisuje pierwszy nagłówek.

Sposób analizowania nagłówków MQ

Do analizowania nagłówków MQ używane jest powiązanie danych nagłówka MQ. Następujące nagłówki są obsługiwane automatycznie:

- MQRFH,
- MQRFH2,
- MQCIH,
- MQIIH.

Nagłówki rozpoczynające się od **MQH** są obsługiwane w inny sposób. Konkretnie pola nagłówka nie są analizowane i pozostają one w postaci nieprzeanalizowanych bajtów.

W przypadku innych nagłówków MQ można napisać niestandardowe powiązania danych nagłówka MQ, aby analizować te nagłówki.

Sposób uzyskiwania dostępu do nagłówków MQ

Dostęp do nagłówków MQ w produkcji może być uzyskiwany na jeden z dwóch sposobów:

- za pośrednictwem obiektu SMO w mediacji,

- za pośrednictwem funkcji API ContextService.

Nagłówki MQ są reprezentowane wewnętrznie przy użyciu elementu MQHeader obiektu SMO. Element MQHeader jest kontenerem na dane nagłówka, który rozszerza element MQControl, ale zawiera element wartości anyType. Zawiera on strukturę MQMD, element MQControl (informacje sterujące treści komunikatu MQ) oraz listę innych nagłówków MQ.

- Struktura MQMD reprezentuje treść opisu komunikatu WebSphere MQ, z wyjątkiem informacji określających strukturę i kodowanie treści.
- Element MQControl zawiera informacje określające strukturę i kodowanie treści komunikatu.
- Element MQHeaders zawiera listę obiektów MQHeader.

Łańcuch nagłówka MQ jest rozwinięty, a więc każdy nagłówek MQ w obiekcie SMO przynosi własne informacje sterujące (identyfikator CCSID, kodowanie i format). Nagłówki można w łatwy sposób dodawać lub usuwać, bez zmiany innych danych nagłówków.

Ustawianie pól w strukturze MQMD

Strukturę MQMD można zaktualizować przy użyciu funkcji API Context lub za pośrednictwem obiektu SMO w mediacji. Do wychodzącego komunikatu MQ automatycznie propagowane są następujące pola:

- Encoding,
- CodedCharacterSet,
- Format,
- Report,
- Expiry,
- Feedback,
- Priority,
- Persistence,
- CorrelId,
- MsgFlags.

Aby propagować następujące właściwości do wychodzącego komunikatu MQ, należy skonfigurować powiązanie MQ importu lub eksportu:

MsgID

Opcję **Identyfikator komunikatu żądania** należy ustawić na wartość kopii z komunikatu SCA.

MsgType

Należy anulować zaznaczenie pola wyboru **Ustaw typ komunikatu na MQMT_DATAGRAM lub MQMT_REQUEST w przypadku operacji żądanie-odpowiedź**.

ReplyToQ

Należy anulować zaznaczenie pola wyboru **Nadpisz kolejkę odpowiedzi komunikatu żądania**.

ReplyToQMgr

Należy anulować zaznaczenie pola wyboru **Nadpisz kolejkę odpowiedzi komunikatu żądania**.

Od wersji 7.0 produktu pola kontekstu można nadpisywać przy użyciu właściwości definiowanej w definicji miejsca docelowego JNDI. Aby propagować następujące pola do wychodzącego komunikatu MQ, należy ustawić właściwość definiowaną MDCTX na wartość SET_IDENTITY_CONTEXT dla miejsca docelowego wysyłania:

- UserIdentifier,
- AppIdentityData.

Aby propagować następujące właściwości do wychodzącego komunikatu MQ, należy ustawić właściwość definiowaną MDCTX na wartość SET_ALL_CONTEXT dla miejsca docelowego wysyłania:

- UserIdentifier,
- ApplIdentityData,
- PutApplType,
- PutApplName,
- ApplOriginData.

Niektóre pola nie są propagowane do wychodzącego komunikatu MQ. Następujące pola są nadpisywane podczas wysyłania komunikatu:

- BackoutCount,
- AccountingToken,
- PutDate,
- PutTime,
- Przesunięcie
- OriginalLength.

Statyczne dodawanie nagłówka MQCIH w wiązaniu WebSphere MQ:

Program IBM Business Process Manager obsługuje statyczne dodawanie informacji nagłówka MQCIH bez stosowania modułu mediacji.

Są różne sposoby dodawania informacji nagłówka MQCIH do komunikatu (na przykład wykorzystanie operacji podstawowej mediacji Header Setter). Czasami może być użyteczne dodanie informacji nagłówka statycznie, bez wykorzystywania dodatkowego modułu mediacji. Statyczny nagłówek zawierający nazwę programu CICS, identyfikator transakcji i inne szczegóły nagłówka formatu danych, można zdefiniować i dodać jako część powiązania WebSphere MQ.

Aby można było statycznie dodać informacje nagłówka MQCIH, programy WebSphere MQ, MQ CICS Bridge i CICS muszą być odpowiednio skonfigurowane.

Do skonfigurowania importu WebSphere MQ z wartościami statycznymi, które są wymagane dla informacji nagłówka MQCIH, można użyć programu Integration Designer.

Gdy komunikat dotrze i jest przetwarzany przez import WebSphere MQ, następuje sprawdzenie, czy informacje nagłówka MQCIH znajdują się już w komunikacie. Jeśli nagłówek MQCIH już jest, używane są wartości statyczne zdefiniowane w imporcie WebSphere MQ, które zastępują odpowiadające im wartości dynamiczne z komunikatu. Jeśli nagłówka MQCIH nie ma, jest on tworzony w komunikacie i dodawane są do niego statyczne wartości zdefiniowane w imporcie WebSphere MQ.

Statyczne wartości zdefiniowane w imporcie WebSphere MQ odnoszą się do konkretnej metody. Dla różnych metod w tym samym imporcie WebSphere MQ można podać różne statyczne wartości nagłówka MQCIH.

To ułatwienie nie jest używane do udostępniania wartości domyślnych, jeśli nagłówek MQCIH nie zawiera konkretnych informacji nagłówka, ponieważ wartość statyczna zdefiniowana w imporcie WebSphere MQ zastępuje odpowiadającą jej wartość w komunikacie przychodzącym.

Klienci zewnętrzne:

Produkt IBM Business Process Manager może wysyłać komunikaty do klientów zewnętrznych i odbierać komunikaty od nich za pośrednictwem powiązań WebSphere MQ.

Klient zewnętrzny (na przykład portal WWW lub system EIS) może wysyłać komunikat do komponentu SCA w aplikacji za pośrednictwem eksportu lub może być wywoływany przez komponent SCA w aplikacji za pośrednictwem importu.

Powiązanie eksportu WebSphere MQ wdraża komponenty bean sterowane komunikatami (Message Driven Bean - MDB) w celu nasłuchiwania żądań przychodzących do miejsca docelowego receive (odbioru) określonego w powiązaniu eksportu. Miejsce docelowe określone w polu send służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołana aplikacja udostępnia odpowiedź. Dzięki temu klient zewnętrzny może wywoływać aplikacje za pośrednictwem powiązania eksportu.

Importy WebSphere MQ są wiązane z klientami zewnętrznymi i mogą dostarczać do nich komunikaty. Komunikat może wymagać odpowiedzi od klienta zewnętrznego, ale może też nie wymagać tej odpowiedzi.

Więcej informacji na temat interakcji z klientami zewnętrznymi przy użyciu produktu WebSphere MQ można znaleźć w Centrum informacyjnym produktu WebSphere MQ.

Rozwiązywanie problemów z powiązaniem produktu WebSphere MQ:

Istnieje możliwość diagnozowania i naprawiania uszkodzeń oraz warunków niepowodzenia związane z powiązaniem produktu WebSphere MQ.

Podstawowe warunki niepowodzenia

Podstawowe warunki niepowodzenia powiązań produktu WebSphere MQ są określane przez semantykę transakcyjną, przez konfigurację produktu WebSphere MQ lub przez odwołanie do aktualnego działania innych komponentów. Istnieją następujące podstawowe warunki niepowodzenia:

- Niepowodzenie nawiązania połączenia z menedżerem kolejek lub z kolejką produktu WebSphere MQ.
Niepowodzenie nawiązania połączenia z produktem WebSphere MQ w celu odbierania komunikatów powoduje, że nie można uruchomić portu nasłuchiwania MDB. Ten warunek zostaje zarejestrowany w dzienniku produktu WebSphere Application Server. Trwałe komunikaty pozostaną w kolejce produktu WebSphere MQ do czasu ich pomyślnego pobrania (lub do utraty ważności określonej przez produkt WebSphere MQ).
Niepowodzenie nawiązania połączenia z produktem WebSphere MQ w celu wysyłania komunikatów wychodzących powoduje wycofanie transakcji sterującej wysyłaniem.
- Niepowodzenie analizy komunikatu przychodzącego lub budowy komunikatu wychodzącego.
Niepowodzenie w powiązaniu danych powoduje wycofanie transakcji sterującej pracą.
- Niepowodzenie wysłania komunikatu wychodzącego.
Niepowodzenie wysłania komunikatu powoduje wycofanie odpowiedniej transakcji.
- Wiele komunikatów odpowiedzi lub nieoczekiwane komunikaty odpowiedzi.
Element importowany oczekuje tylko jednego komunikatu odpowiedzi na każdy komunikat żądania. Jeśli nadejdzie więcej niż jedna odpowiedź lub nadejdzie odpowiedź spóźniona (czas utraty ważności odpowiedzi SCA minął), zgłoszony zostanie wyjątek środowiska wykonawczego usługi. Transakcja zostanie wycofana, a komunikat odpowiedzi zostanie usunięty z kolejki lub obsłużony przez menedżer zdarzeń zakończonych niepowodzeniem.

Scenariusze niepoprawnego użycia: porównanie z powiązaniem JMS produktu WebSphere MQ

Elementy importowane i eksportowane produktu WebSphere MQ zostały zaprojektowane głównie na potrzeby współpracy z rodzimymi aplikacjami produktu WebSphere MQ i ujawniają mediacjom pełną treść komunikatu produktu WebSphere MQ. Powiązanie JMS produktu WebSphere MQ zostało jednak zaprojektowane z myślą o współpracy z aplikacjami JMS wdrożonymi z produktem WebSphere MQ, który ujawnia komunikaty zgodnie z modelem komunikatów JMS.

W następujących scenariuszach należy stosować powiązanie JMS produktu WebSphere MQ zamiast powiązań produktu WebSphere MQ:

- Wywoływanie komponentu bean sterowanego komunikatami JMS (MDB) z modułu SCA, gdy komponent MDB jest wdrożony w dostawcy JMS produktu WebSphere MQ. Należy użyć elementu importowanego JMS produktu WebSphere MQ.

- Umożliwienie wywołania modułu SCA z serwletu komponentów Java EE lub z komponentu EJB za pomocą usług JMS. Należy użyć elementu eksportowanego JMS produktu WebSphere MQ.
- Mediowanie treści komunikatu JMS MapMessage w trakcie przejścia przez produkt WebSphere MQ. Należy użyć elementu eksportowanego i elementu importowanego JMS produktu WebSphere MQ w połączeniu z odpowiednim powiązaniem danych.

Są takie sytuacje, w których powiązanie produktu WebSphere MQ i powiązanie JMS produktu WebSphere MQ JMS powinny współpracować. W szczególności podczas łączenia aplikacji Java EE i innych niż Java EE w produkcie WebSphere MQ należy używać elementu eksportowanego produktu WebSphere MQ oraz elementu importowanego JMS produktu WebSphere MQ (lub odwrotnie) w połączeniu z odpowiednimi powiązaniem danych lub modułami mediacji (bądź dwoma tymi rodzajami obiektów).

Niedostarczone komunikaty

Gdy produkt WebSphere MQ nie może dostarczyć komunikatu do właściwego miejsca docelowego (na przykład z powodu błędów konfiguracji), wysyła go do wyznaczonej kolejki niedostarczonych komunikatów.

Jednocześnie dodaje na początku treści komunikatu nagłówek sygnalizujący niedostarczenie. Ten nagłówek zawiera m.in. informacje o przyczynach niepowodzenia oraz pierwotnym miejscu docelowym.

Komunikaty SCA oparte na produkcie MQ nie są wyświetlane w menedżerze zdarzeń zakończonych niepowodzeniem

Jeśli komunikaty SCA są wynikiem niepowodzenia interakcji produktu WebSphere MQ, powinny znaleźć się w menedżerze zdarzeń zakończonych niepowodzeniem. Jeśli takie komunikaty nie pojawiają się w menedżerze zdarzeń zakończonych niepowodzeniem, należy sprawdzić, czy w bazowym miejscu docelowym produktu WebSphere MQ maksymalną liczbę nieudanych dostaw ustawiono na wartość większą niż 1. Ustawienie wartości 2 lub większej umożliwia interakcję z menedżerem zdarzeń zakończonych niepowodzeniem w trakcie wywołań SCA dotyczących powiązań produktu WebSphere MQ.

Zdarzenia zakończone niepowodzeniem produktu MQ są odtwarzane do niewłaściwego menedżera kolejek

Jeśli na potrzeby połączeń wychodzących ma być używana predefiniowana fabryka połączeń, właściwości połączenia muszą być zgodne z właściwościami zdefiniowanymi w specyfikacji aktywowania używanej na potrzeby połączeń przychodzących.

Predefiniowana fabryka połączeń jest używana do tworzenia połączenia podczas odtwarzania zdarzenia zakończonego niepowodzeniem i dlatego w jej konfiguracji musi być uwzględnione użycie tego samego menedżera kolejek, z którego odebrano pierwotnie komunikat.

Obsługa wyjątków:

Sposób skonfigurowania powiązania określa sposób obsługi wyjątków wywołanych przez procedury obsługi danych lub powiązania danych. Na działanie systemu podczas zgłoszenia takiego wyjątku dodatkowy wpływ ma rodzaj przepływu mediacji.

Jeśli procedura obsługi danych lub powiązanie danych zostanie wywołane przez powiązanie użytkownika, może wystąpić wiele różnych problemów. Procedura obsługi danych może na przykład odebrać komunikat z uszkodzonym ładunkiem lub może próbować odczytać komunikat w niepoprawnym formacie.

Sposób obsługi takiego wyjątku przez powiązanie użytkownika jest określany na podstawie tego, w jaki sposób zaimplementowano procedurę obsługi danych lub powiązanie danych. Zalecanym sposobem utworzenia powiązania danych jest zaprojektowanie go tak, aby zgłaszało wyjątek **DataBindingException**.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, każdy wyjątek procedury obsługi danych jest opakowany w wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

W przypadku wywołania wyjątku czasu wykonywania, między innymi wyjątku **DataBindingException**, mają miejsce następujące działania:

- Jeśli przepływ mediacji jest skonfigurowany jako transakcyjny, komunikat JMS jest domyślnie zapisywany w menedżerze zdarzeń zakończonych niepowodzeniem w celu ręcznego powtórzenia lub usunięcia.

Uwaga: Użytkownik może zmienić tryb odtwarzania dla powiązania tak, aby komunikat był wycofywany zamiast zapisywania go w menedżerze zdarzeń zakończonych niepowodzeniem.

- Jeśli przepływ mediacji nie jest transakcyjny, wyjątek jest rejestrowany, a komunikat zostaje utracony.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, wyjątek procedury obsługi danych jest tworzony wewnątrz wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

Ograniczenia powiązań:

W niniejszej sekcji wymieniono ograniczenia korzystania z powiązań.

Ograniczenia powiązania MQ:

Powiązanie MQ charakteryzuje się pewnymi ograniczeniami sposobu używania. Zostały one opisane w niniejszej sekcji.

Brak dystrybucji komunikatów przy użyciu publikowania i subskrybowania

Metoda dystrybucji komunikatów oparta na publikowaniu i subskrybowaniu nie jest aktualnie obsługiwana przez powiązanie MQ, mimo że produkt WMQ obsługuje mechanizm publikowania-subskrybowania. Jednak powiązanie MQ JMS obsługuje tę metodę dystrybucji.

Współużytkowane kolejki odbiorcze

W przypadku wielu powiązań eksportu i importu WebSphere MQ oczekiwane jest, że wszystkie komunikaty obecne w skonfigurowanej dla nich kolejce odbiorczej są skierowane do tego eksportu lub importu. Dlatego też powiązania importu i eksportu należy konfigurować, biorąc pod uwagę następujące kwestie:

- Każdy import MQ musi mieć inną kolejkę odbiorczą, ponieważ powiązanie importu MQ zakłada, że wszystkie komunikaty w kolejce odbiorczej są odpowiedziami na żądania wysłane przez to powiązanie. Jeśli kolejka odbiorcza jest współużytkowana przez więcej niż jeden import, odpowiedzi mogą być odebrane przez niepoprawny import i nie uda się ich korelacja z pierwotnym komunikatem żądania.
- Każdy eksport MQ powinien mieć inną kolejkę odbiorczą, ponieważ w przeciwnym razie nie można przewidzieć, który eksport otrzyma konkretny komunikat żądania.
- Importy i eksporty MQ mogą wskazywać tę samą kolejkę nadawczą.

Ograniczenia powiązań JMS, MQ JMS i ogólnych powiązań JMS:

Powiązania JMS i MQ JMS charakteryzują się pewnymi ograniczeniami.

Konsekwencje generowania powiązań domyślnych

Ograniczenia używania powiązań JMS, MQ JMS i ogólnych powiązań JMS zostały omówione w następujących sekcjach:

- Konsekwencje generowania powiązań domyślnych,
- Schemat korelacji odpowiedzi,

- Obsługa języków dwukierunkowych.

Kiedy użytkownik generuje powiązanie, kilka pól jest wypełnianych wartościami domyślnymi, jeśli nie zostaną wprowadzone w nich żadne wartości. W ten sposób tworzona jest na przykład nazwa fabryki połączeń. Jeśli wiadomo, że aplikacja będzie umieszczana na serwerze, a dostęp do niej będzie uzyskiwany zdalnie za pomocą klienta, na etapie tworzenia powiązania należy wprowadzić nazwy JNDI zamiast przyjmować wartości domyślne, ponieważ jest prawdopodobne, że wartości te będą wymagać kontrolowania za pomocą Konsoli administracyjnej w czasie wykonywania.

Jeśli jednak użytkownik zaakceptuje wartości domyślne, a w późniejszym czasie okaże się, że nie można uzyskać dostępu do aplikacji z poziomu klienta zdalnego, można użyć Konsoli administracyjnej do jawnego ustawienia wartości fabryki połączeń. Należy w tym celu znaleźć pole punktów końcowych dostawcy w ustawieniach fabryki połączeń i dodać wartość, na przykład <nazwa_hosta_serwera>:7276 (jeśli wykorzystywany jest domyślny numer portu).

Schemat korelacji odpowiedzi

Jeśli używany jest schemat korelacji odpowiedzi CorrelationID To CorrelationID, który służy do korelowania komunikatów w operacji żądanie-odpowiedź, w komunikacie należy stosować dynamiczny identyfikator korelacji.

Aby utworzyć dynamiczny identyfikator korelacji w module mediacji przy użyciu edytora przepływów mediacji, należy dodać operację podstawową mediacji Odwzorowanie przed importem zawierającym powiązanie JMS. W tym celu należy najpierw otworzyć edytor odwzorowań. Znane nagłówki architektury SCA będą dostępne w komunikacie docelowym. Następnie należy przeciągnąć pole zawierające unikalny identyfikator w komunikacie źródłowym na identyfikator korelacji w nagłówku JMS komunikatu docelowego.

Obsługa języków dwukierunkowych

W czasie wykonywania w nazwach JNDI (Java Naming and Directory Interface) obsługiwane są wyłącznie znaki ASCII.

Współużytkowane kolejki odbiorcze

W przypadku wielu powiązań eksportu i importu oczekiwane jest, że wszystkie komunikaty obecne w skonfigurowanej dla nich kolejce odbiorczej są skierowane do tego eksportu lub importu. Dlatego też powiązania importu i eksportu należy konfigurować, biorąc pod uwagę następujące kwestie:

- Każde powiązanie importu musi mieć inną kolejkę odbiorczą, ponieważ powiązanie importu zakłada, że wszystkie komunikaty w kolejce odbiorczej są odpowiedziami na żądania wysłane przez to powiązanie. Jeśli kolejka odbiorcza jest współużytkowana przez więcej niż jeden import, odpowiedzi mogą być odebrane przez niepoprawny import i nie uda się ich korelacja z pierwotnym komunikatem żądania.
- Każdy eksport powinien mieć inną kolejkę odbiorczą, ponieważ w przeciwnym razie nie można przewidzieć, który eksport otrzyma konkretny komunikat żądania.
- Importy i eksporty mogą wskazywać tę samą kolejkę nadawczą.

Obiekty biznesowe

W branży oprogramowania komputerowego powstało wiele modeli oprogramowania i środowisk, w których *obiekty biznesowe* stanowią naturalną reprezentację danych biznesowych przetwarzanych przez aplikacje.

Ogólnie, obiekty biznesowe:

- są definiowane przy użyciu standardów branżowych,
- w sposób przezroczysty odwzorowują dane na tabele baz danych lub systemy informacji w przedsiębiorstwach,
- obsługują protokoły zdalnych wywołań,
- udostępniają platformę modelu programowania danych na potrzeby programowania aplikacji.

Komponent Process Designer i produkt Integration Designer udostępniają programistom popularny model obiektów biznesowych do reprezentowania różnych rodzajów jednostek biznesowych z różnych domen.

W produkcie Process Designer obiekty biznesowe koncentrują się na reprezentacji typu danych. Podstawowe obiekty biznesowe (typy zmiennych) są udostępniane w systemowych bibliotekach narzędziowych, a użytkownik może stworzyć niestandardowe typy zmiennych nazywane niestandardowymi obiektami biznesowymi. Więcej informacji na ten temat zawiera sekcja Obiekty biznesowe i zmienne.

W komponencie Integration Designer, który jest dostępny tylko w produkcie IBM Business Process Manager Advanced, obiekty biznesowe mogą reprezentować bardziej skomplikowane konstrukcje XSD. W komponencie Integration Designer obiekty biznesowe są blisko spowinowaczone ze schematami XML. Informacje o tym, co należy wziąć pod uwagę podczas integrowania obiektów biznesowych, które zostały zdefiniowane w komponencie Integration Designer przy użyciu obiektów biznesowych zdefiniowanych w produkcie Process Designer zawierają sekcje Zapis lustrzany bibliotek i Nieobsługiwane konstrukcje XML.

W trakcie programowania w komponencie Integration Designer model obiektu biznesowego umożliwia programistom definiowanie obiektów biznesowych jako definicji schematów XML. W czasie wykonywania dane biznesowe zdefiniowane w schematach XML są reprezentowane jako obiekty biznesowe Java. W tym modelu obiekty biznesowe są w mniejszym stopniu oparte na wcześniejszych wersjach roboczych specyfikacji SDO (Service Data Object) i udostępniają kompletny zestaw interfejsów aplikacyjnych modelu programowania wymaganych do manipulowania danymi biznesowymi. Następujące tematy podrzędne zawierają dokładniejsze informacje na temat używania obiektów biznesowych w komponencie Integration Designer.

Definiowanie obiektów biznesowych

Obiekty biznesowe są definiowane przy użyciu edytora obiektów biznesowych w produkcie Integration Designer. Edytor obiektów biznesowych przechowuje obiekty biznesowe jako definicje schematów XML.

Używanie schematu XML do definiowania obiektów biznesowych wiąże się z kilkoma korzyściami:

- Schematy XML udostępniają model definicji danych oparty na standardach oraz platformę współdziałania między odmiennymi, heterogenicznymi systemami i aplikacjami. Schematy XML są używane w połączeniu z językiem WSDL (Web Services Description Language), udostępniając oparte na standardach kontrakty interfejsu między komponentami, aplikacjami i systemami.
- Schematy XML określają rozbudowany model definicji danych służący do reprezentowania danych biznesowych. Ten model zawiera między innymi typy złożone, typy proste, typy zdefiniowane przez użytkownika, dziedziczenie typów oraz licznosc.
- Obiekty biznesowe mogą być definiowane na podstawie interfejsów biznesowych oraz danych zdefiniowanych w języku WSDL (Web Services Description Language), a także na podstawie schematów XML pochodzących od organizacji nadzorujących standardy branżowe lub z innych systemów i aplikacji. Integration Designer może bezpośrednio importować te obiekty biznesowe.

Integration Designer udostępnia również obsługę wykrywania danych biznesowych w bazach danych oraz korporacyjnych systemach informacyjnych, a także umożliwia generowanie definicji obiektów biznesowych w postaci standardowych schematów XML dla tych danych biznesowych. Obiekty biznesowe wygenerowane w ten sposób są często nazywane *obektami biznesowymi specyficznymi dla aplikacji*, ponieważ odzwierciedlają strukturę danych biznesowych zdefiniowanych w korporacyjnym systemie informacyjnym.

Jeśli proces manipuluje danymi z wielu różnych systemów informacyjnych, warto przetransformować odmienne reprezentacje danych biznesowych (na przykład Klient_EIS1 i Klient_EIS2 lub Zamówienie_EIS1 i Zamówienie_EIS2) na pojedynczą reprezentację kanoniczną (na przykład Klient lub Zamówienie). Reprezentacja kanoniczna jest często nazywana *ogólnym obiektem biznesowym*.

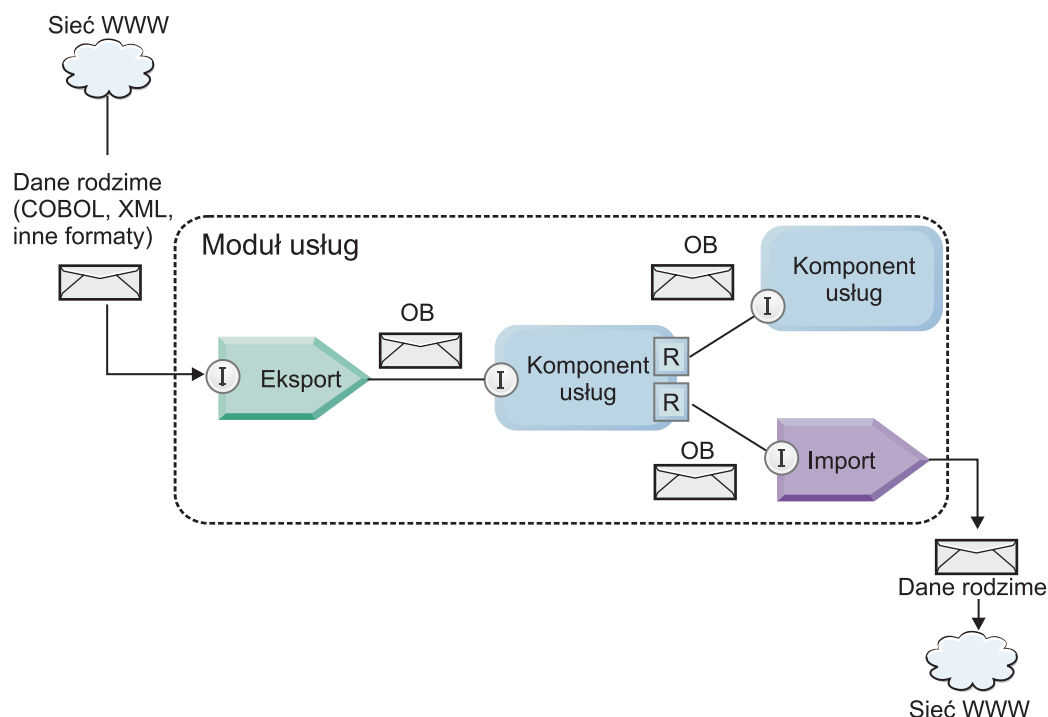
Definicje obiektów biznesowych, szczególnie w przypadku ogólnych obiektów biznesowych, są często używane przez więcej niż jedną aplikację. Aby umożliwić to ponowne wykorzystywanie, produkt Integration Designer pozwala tworzyć obiekty biznesowe w bibliotekach, które można następnie powiązać z wieloma modułami aplikacji.

Zadania dla usług udostępnianych przez moduł aplikacji SCA (Service Component Architecture) i konsumowanych przez ten moduł, jak również zadania używane do tworzenia komponentów w module aplikacji, są definiowane przy użyciu języka WSDL (Web Services Description Language). W zadaniu dokument WSDL może reprezentować zarówno operacje, jak i obiekty biznesowe, które są definiowane przez schemat XML w celu reprezentowania danych biznesowych.

Praca z obiektami biznesowymi

Architektura SCA (Service Component Architecture) udostępnia środowisko do definiowania modułu aplikacji, usługi do jego udostępniania i wykorzystywania oraz zestaw komponentów udostępniających jego logikę biznesową. Obiekty biznesowe pełnią w aplikacji ważną rolę: służą do definiowania danych biznesowych używanych do opisu zadań usług i komponentów oraz danych biznesowych przetwarzanych przez komponenty.

Na następującym diagramie przedstawiono moduł aplikacji SCA i pokazano wiele miejsc, w których programista pracuje z obiektami biznesowymi.



Rysunek 41. Obiekty biznesowe jako reprezentacja danych przekazywanych między usługami w aplikacji

Uwaga: W tym temacie opisano, w jaki sposób obiekty biznesowe są używane w modułach aplikacji SCA. Jeśli programista używa interfejsów Java, w modułach aplikacji SCA może przetwarzać również obiekty Java.

Model programowania obiektów biznesowych

Model programowania obiektów biznesowych składa się z zestawu interfejsów Java, które reprezentują:

- definicję obiektu biznesowego i dane instancji,
- zestaw usług do obsługi operacji na obiektach biznesowych.

Definicje typów obiektów biznesowych są reprezentowane przez interfejsy `commonj.sdo.Type` i `commonj.sdo.Property`. Model programowania obiektów biznesowych udostępnia zestaw reguł odwzorowywania informacji zawartych w typach złożonych schematu XML na interfejs `Type` oraz odwzorowywania wszystkich elementów z definicji typu złożonego na interfejs `Property`.

Instancje obiektów biznesowych są reprezentowane przez interfejs `commonj.sdo.DataObject`. Model programowania obiektów biznesowych jest beztypowy; oznacza to, że ten sam interfejs `commonj.sdo.DataObject` może być używany do reprezentowania różnych definicji obiektów biznesowych, takich jak Klient i Zamówienie. O tym, które właściwości mogą być ustawiane i pobierane z każdego obiektu biznesowego, decyduje typ informacji zdefiniowanych w schemacie XML powiązany z każdym obiektem biznesowym.

Zachowanie modelu programowania obiektów biznesowych jest oparte na specyfikacji Service Data Object 2.1. Dodatkowe informacje można znaleźć w specyfikacji SDO 2.1 for Java oraz w podręcznikach i w dokumentacji języka Java pod adresem: <http://www.oasis-openca.org/>.

Usługi obiektów biznesowych obsługują operacje na różnych etapach cyklu życia obiektów biznesowych, takie jak tworzenie, porównywanie, analizowanie czy serializacja obiektów biznesowych.

Specyfikację modelu programowania obiektów biznesowych można znaleźć w sekcjach Programowanie korzystania z usług obiektów biznesowych i Dokumentacja wygenerowanych interfejsów API i interfejsów SPI dla obiektów biznesowych.

Powiązania, powiązania danych i procedury obsługi danych

Rys. 41 na stronie 151 przedstawia, w jaki sposób dane biznesowe używane podczas wywoływania usług udostępnianych przez moduły aplikacji SCA są transformowane w obiekty biznesowe, aby umożliwić manipulowanie nimi w komponentach SCA. Podobnie, obiekty biznesowe przetwarzane w komponentach SCA są przekształcane w format danych wymagany przez usługi zewnętrzne.

W niektórych przypadkach, na przykład podczas wiązania usługi Web Service, powiązanie używane w usługach eksportu i importu automatycznie transformuje dane do odpowiedniego formatu. W innych przypadkach, na przykład podczas wiązania JMS, programiści mogą udostępnić powiązanie danych lub procedurę obsługi danych, która przekształca formaty inne niż rodzime w obiekty biznesowe reprezentowane przez interfejs `DataObject`.

Więcej informacji na temat programowania powiązań danych i procedur obsługi danych można znaleźć w sekcjach “Procedury obsługi danych” na stronie 57 oraz “Powiązania danych” na stronie 59.

Komponenty

Komponenty SCA zawierają definicje zadań usług udostępniania i wykorzystywania przy użyciu języka WSDL (Web Services Description Language) i schematu XML. Dane biznesowe przekazywane w środowisku SCA między komponentami są reprezentowane przez obiekty biznesowe przy użyciu interfejsu `DataObject`. W środowisku SCA następuje sprawdzenie, czy typy obiektów biznesowych są zgodne z zadaniem interfejsu zdefiniowanym w wywołanym komponentcie.

Abstrakcje modelu programowania do manipulowania obiektami biznesowymi są różne w różnych komponentach. Komponent POJO i operacja podstawowa Custom w komponentcie przepływu mediacji umożliwiają bezpośrednie manipulowanie obiektami biznesowymi, zezwalając na używanie języka Java bezpośrednio w usługach i interfejsach programowania obiektów biznesowych. Większość komponentów udostępnia abstrakcje wyższego poziomu do manipulowania obiektami biznesowymi, ale dostępne są także fragmenty kodu w języku Java do definiowania niestandardowego zachowania usług i interfejsów obiektów biznesowych.

Obiekty biznesowe można transformować przy użyciu kombinacji komponentów Interface Flow Mediation i Business Object Map albo przy użyciu komponentu przepływu mediacji i jego operacji podstawowej XML Map. Te możliwości transformacji obiektów biznesowych są użyteczne do przekształcania obiektów biznesowych specyficznych dla aplikacji w ogólne obiekty biznesowe i odwrotnie.

Specjalne obiekty biznesowe

Obiekty SMO i grafy biznesowe to dwa specjalne typy obiektów biznesowych, używane w aplikacjach do specjalnych celów.

Obiekt SMO

Obiekt SMO (Service Message Object) jest to specjalny obiekt biznesowy używany przez komponenty przepływu mediacji do reprezentowania kolekcji danych powiązanych z wywołaniem usługi.

Obiekt SMO ma stałą strukturę najwyższego poziomu, która składa się z nagłówek, kontekstu, treści i załączników (jeśli istnieją).

- W nagłówkach są przechowywane informacje związane z wywołaniem usługi poprzez konkretny protokół lub powiązanie. Przykładowo są to nagłówki SOAP i nagłówki JMS.
- Dane kontekstowe to dodatkowe informacje logicznie powiązane z wywołaniem, które jest przetwarzane przez komponent przepływu mediacji. Zwykle te informacje nie są częścią danych aplikacji wysyłanych lub otrzymywanych przez klienty.
- Treść obiektu SMO to ładunek danych biznesowych, który reprezentuje zasadniczy komunikat aplikacji lub dane wywołania w formie standardowego obiektu biznesowego.

Obiekt SMO może zawierać również załączniki umożliwiające wywoływanie usług Web Services przy użyciu nagłówek SOAP z załącznikami.

W przepływach mediacji wykonywane są takie zadania, jak żądania przepływu i transformacja danych, a obiekt SMO zapewnia złożony widok nagłówek i treści ładunku w jednej ujednoczonej strukturze.

Graf biznesowy

Graf biznesowy to specjalny obiekt biznesowy używany do obsługi synchronizacji danych w scenariuszach integracji.

Rozważmy przykład, w którym w dwóch korporacyjnych systemach informacyjnych istnieją reprezentacje konkretnej kolejności. Gdy kolejność w jednym systemie ulegnie zmianie, do drugiego systemu może zostać wysłany komunikat informujący o potrzebie synchronizacji kolejności danych. Grafy biznesowe obsługują przesyłanie do drugiego systemu zmienionej porcji danych oraz informacji podsumowujących zmiany i definiujących ich typ.

W tym przykładzie graf biznesowy Kolejność mógłby przekazać do innego systemu informację, że jeden z wierszy w kolejności został usunięty i że właściwość wysyłania danych została zaktualizowana.

Grafy biznesowe można łatwo dodawać do istniejących obiektów biznesowych, korzystając z programu Integration Designer. Występują one najczęściej w scenariuszach, w których są używane adaptory WebSphere, oraz w scenariuszach obsługi migracji aplikacji serwera WebSphere InterChange Server.

Tryb analizowania obiektu biznesowego

Produkt Integration Designer udostępnia właściwość dla modułów lub bibliotek, której można użyć do zmiany trybu analizowania XML dla obiektów biznesowych na przyspieszony lub opóźniony.

- Jeśli wybrana jest opcja *przyspieszony*, strumienie bajtów XML są analizowane w trybie przyspieszonym w celu utworzenia obiektu biznesowego.
- Jeśli wybrana jest opcja *opóźniony*, obiekt biznesowy jest tworzony w normalny sposób, ale rzeczywiste analizowanie strumienia bajtów XML jest odraczane i realizowane częściowo dopiero podczas uzyskiwania dostępu do właściwości obiektu biznesowego.

W obu trybach analizowania XML dane inne niż XML są zawsze analizowane w trybie przyspieszonym w celu utworzenia obiektu biznesowego.

Uwagi na temat wybierania trybu analizowania obiektu biznesowego:

Tryb analizowania obiektu biznesowego określa sposób analizowania danych XML w czasie wykonywania. Tryb analizowania obiektu biznesowego jest definiowany w module lub w bibliotece podczas ich tworzenia. Tryb analizowania dla modułu lub biblioteki można zmienić, ale należy zdawać sobie sprawę z konsekwencji.

Tryb analizowania obiektu biznesowego jest ustawiany na poziomie modułu i biblioteki. Moduły, które zostały utworzone w wersji produktu IBM Integration Designer wcześniejszej niż wersja 7, będą uruchamiane w przyspieszonym trybie analizowania bez konieczności wprowadzania zmian. Moduły i biblioteki utworzone w produkcie IBM Integration Designer w wersji 7 lub późniejszej domyślnie otrzymają najodpowiedniejszy tryb analizowania, w zależności od kilku czynników takich jak tryb analizowania istniejących projektów w obszarze roboczym lub tryb analizowania projektów zależnych albo innych projektów zawartych w tym samym rozwiązaniu itp. Tryb analizowania obiektu biznesowego dla modułu lub biblioteki można zmienić tak, aby był odpowiedni dla implementacji, ale należy uwzględnić następujące uwagi.

Uwagi

- Opóźniony tryb analizowania obiektu biznesowego przetwarza dane XML szybciej. Występują jednak różnice dotyczące zgodności między trybem przyspieszonym i trybem opóźnionym, z których należy zdawać sobie sprawę przed dokonaniem zmiany konfiguracji modułu lub biblioteki. Te różnice będą miały wpływ na działanie modułów w czasie wykonywania. Informacje o tym, który tryb analizowania jest optymalny dla danej aplikacji, zawiera temat **Zalety stosowania opóźnionego (zamiast przyspieszonego) trybu analizowania** w sekcji stron pokrewnych.
- Moduł można skonfigurować do działania tylko w jednym trybie analizowania. Biblioteki można skonfigurować tak, aby obsługiwały jeden z dostępnych trybów analizowania lub oba z nich. Do biblioteki skonfigurowanej tak, aby obsługiwała oba tryby analizowania może odwoływać się zarówno moduł używający przyspieszonego trybu analizowania, jak i moduł używający opóźnionego trybu analizowania. Tryb analizowania biblioteki stosowany w czasie wykonywania jest określany przez moduły odwołujące się do tej biblioteki. W czasie wykonywania moduł deklaruje swój tryb analizowania. Ten tryb analizowania jest następnie używany przez moduł oraz wszystkie biblioteki używane przez ten moduł.
- Moduły i biblioteki skonfigurowane dla różnych trybów analizowania są zgodne w następujących przypadkach:
 - Moduły i biblioteki skonfigurowane z opóźnionym trybem analizowania są zgodne z bibliotekami, które używają opóźnionego trybu analizowania lub zarówno przyspieszonego jak i opóźnionego trybu analizowania.
 - Moduły i biblioteki skonfigurowane z przyspieszonym trybem analizowania są zgodne z bibliotekami, które używają przyspieszonego trybu analizowania lub zarówno przyspieszonego jak i opóźnionego trybu analizowania.
 - Biblioteki skonfigurowane z opóźnionym i przyspieszonym trybem analizowania są zgodne tylko z bibliotekami, które używają zarówno opóźnionego, jak i przyspieszonego trybu analizowania.
- Dla współdziałających modułów komunikujących się przy użyciu powiązania SCA należy użyć tego samego trybu analizowania. Jeśli moduły komunikują się przy użyciu różnych trybów analizowania, mogą wystąpić problemy związane z wydajnością.

Pojęcia pokrewne:

“Zalety stosowania opóźnionego (zamiast przyspieszonego) trybu analizowania”

W jednych aplikacjach za korzystniejszy uznaje się opóźniony tryb analizowania XML, w innych lepszą wydajność zapewnia stosowanie przyspieszonego trybu analizowania. Zaleca się wykonanie testu porównawczego obu trybów analizowania i na tej podstawie określenie, który tryb jest bardziej odpowiedni dla konkretnej aplikacji.

Zalety stosowania opóźnionego (zamiast przyspieszonego) trybu analizowania:

W jednych aplikacjach za korzystniejszy uznaje się opóźniony tryb analizowania XML, w innych lepszą wydajność zapewnia stosowanie przyspieszonego trybu analizowania. Zaleca się wykonanie testu porównawczego obu trybów analizowania i na tej podstawie określenie, który tryb jest bardziej odpowiedni dla konkretnej aplikacji.

Aplikacje, w których są analizowane duże strumienie danych XML, prawdopodobnie będą miały lepszą wydajność, gdy będzie używany opóźniony tryb analizowania XML. Poprawa wydajności będzie większa, gdy strumień bajtów XML będzie wzrastał, a ilość danych w strumieniu bajtów, do którego aplikacja uzyskuje dostęp, będzie maleć.

Następujące aplikacje prawdopodobnie będą osiągały lepszą wydajność przy użyciu przyspieszonego trybu analizowania:

- Aplikacje, w których są analizowane strumienie danych innych niż XML.
- Aplikacje, które korzystają z komunikatów utworzonych przy użyciu usługi BOFactory.

- Aplikacje, w których są analizowane bardzo małe komunikaty XML.

Odsyłacze pokrewne:

“Uwagi na temat wybierania trybu analizowania obiektu biznesowego” na stronie 153

Tryb analizowania obiektu biznesowego określa sposób analizowania danych XML w czasie wykonywania. Tryb analizowania obiektu biznesowego jest definiowany w module lub w bibliotece podczas ich tworzenia. Tryb analizowania dla modułu lub biblioteki można zmienić, ale należy zdawać sobie sprawę z konsekwencji.

Uwagi na temat migrowania i programowania aplikacji:

Podczas konfigurowania aplikacji, która została napisana pod kątem stosowania przyspieszonego trybu analizowania, w przypadku korzystania z opóźnionego trybu analizowania lub podczas planowania przełączania aplikacji między opóźnionym i przyspieszonym trybem analizowania należy pamiętać o różnicach między trybami oraz o zagadnieniach związanych z przełączaniem trybów.

Obsługa błędów

Jeśli strumień bajtów XML jest uszkodzony, wystąpią wyjątki analizowania.

- W przyspieszonym trybie analizowania XML wyjątki są generowane w trakcie analizowania obiektu biznesowego z przychodzącego strumienia danych XML.
- Jeśli zostanie skonfigurowany opóźniony tryb analizowania XML, wyjątki analizowania wystąpią z pewnym opóźnieniem — po uzyskaniu dostępu do właściwości obiektów biznesowych i po zakończeniu analizowania błędnej porcji danych XML.

W celu obsługi błędnych danych XML należy wybrać jedną z następujących opcji:

- Wdrożenie magistrali komunikacyjnej na krawędziach, aby sprawdzać poprawność przychodzących plików XML.
- Utworzenie logiki opóźnionego wykrywania błędów w punkcie, w którym są dostępne właściwości obiektów biznesowych.

Stosy wyjątków i komunikaty

Ponieważ przyspieszony i opóźniony tryb analizowania XML mają różne implementacje bazowe, stosy wywołań zgłaszane przez usługi i interfejsy programistyczne obiektów biznesowych mają wprawdzie taką samą nazwę klasy wyjątków, ale mogą nie zawierać takiego samego komunikatu o wyjątku lub opakowanego zestawu klas wyjątków przeznaczonych dla konkretnej implementacji.

Format serializacji XML

W opóźnionym trybie analizowania XML optymalizowanie wydajności podczas serializacji polega na próbach kopiowania niezmiennych fragmentów XML z przychodzącego strumienia bajtów do strumienia wychodzącego. Powoduje to wzrost wydajności, ale format serializacji wychodzącego strumienia bajtów XML może zostać zmieniony, jeśli cały obiekt biznesowy został zaktualizowany w opóźnionym trybie analizowania XML lub gdy był uruchomiony w przyspieszonym trybie analizowania XML.

Nawet jeśli format serializacji XML nie będzie dokładnie taki sam pod względem syntaktycznym, wartość semantyczna udostępniana przez obiekt biznesowy będzie dokładnie taka sama, niezależnie od użytych trybów analizowania, a strumień XML można bezpiecznie przekazywać między aplikacjami używającymi różnych trybów analizowania.

Analizator poprawności instancji obiektów biznesowych

Analizator poprawności instancji obiektów biznesowych dla opóźnionego trybu analizowania XML udostępnia dokładniejsze sprawdzanie poprawności obiektów biznesowych, a szczególnie sprawdzanie poprawności aspektów wartości właściwości. Dzięki temu analizator poprawności instancji obiektów biznesowych dla opóźnionego trybu analizowania wychwytuje dodatkowe problemy w porównaniu z analizatorem dla przyspieszonego trybu analizowania oraz udostępnia bardziej szczegółowe komunikaty o błędach.

Odwzorowania w wersji 602 języka XML

Przepływy mediacji, które zostały pierwotnie utworzone w programie WebSphere Integration Developer w wersji wcześniejszej niż 6.1, mogą zawierać operacje podstawowe Odwzorowanie używające odwzorowania lub arkusza stylów, których nie można bezpośrednio uruchamiać w opóźnionym trybie analizowania XML. Podczas migrowania aplikacji, aby umożliwić stosowanie opóźnionego trybu analizowania XML, pliki odwzorowania powiązane z operacjami podstawowymi Odwzorowanie mogą zostać automatycznie zaktualizowane przez kreator migracji do działania w nowym trybie. Jeśli jednak operacja podstawowa Odwzorowanie odwołuje się bezpośrednio do arkusza stylów, który był edytowany ręcznie, arkusz stylów nie zostanie poddany migracji i nie będzie można go uruchomić w opóźnionym trybie analizowania XML.

Prywatne niepublikowane interfejsy API

Jeśli w aplikacji są wykorzystywane niepublikowane, prywatne interfejsy programistyczne obiektów biznesowych przeznaczone dla konkretnej implementacji, wówczas prawdopodobnie po przełączeniu trybu analizowania kompilowanie aplikacji nie powiedzie się. W przyspieszonym trybie analizowania te prywatne interfejsy są zwykle klasami implementacji obiektów biznesowych zdefiniowanymi w środowisku EMF (Eclipse Modeling Framework).

We wszystkich przypadkach usilnie zaleca się usunięcie prywatnych interfejsów API z aplikacji.

Interfejsy API dla obiektów SMO w środowisku EMF

Komponent mediacji w programie IBM Integration Designer udostępnia możliwość manipulowania zawartością komunikatów przy użyciu klas w języku Java oraz interfejsów dostarczanych w pakiecie `com.ibm.websphere.sibx.smobo`. W opóźnionym trybie analizowania XML można używać interfejsów w języku Java z pakietu `com.ibm.websphere.sibx.smobo`, jednak metody, które odwołują się bezpośrednio do interfejsów i klas EMF (Eclipse Modeling Framework) lub które są dziedziczone z interfejsów EMF, prawdopodobnie nie powiodą się.

W opóźnionym trybie analizowania XML obiektu `ServiceMessageObject` i jego treści nie można rzutować na obiekty EMF.

Usługa BOMode

Usługa BOMode służy do określania, czy bieżącym trybem analizowania XML jest tryb opóźniony czy przyspieszony.

Migracja

Wszystkie aplikacje w wersjach wcześniejszych niż wersja 7.0.0.0 działają w przyspieszonym trybie analizowania XML. Jeśli zostały poddane migracji podczas wykonywania przez narzędzia do migracji BPM w czasie wykonywania, w dalszym ciągu będą działać w przyspieszonym trybie analizowania XML.

Aby można było skonfigurować aplikację w wersji wcześniejszej niż wersja 7.0.0.0 do używania opóźnionego trybu analizowania XML, najpierw należy użyć programu Integration Designer i poddać migracji artefakty aplikacji. Po zakończeniu migracji można skonfigurować aplikację do używania opóźnionego trybu analizowania XML.

Informacje o migrowaniu artefaktów w programie Integration Designer zawiera temat Migrowanie artefaktów źródłowych, a informacje o ustawianiu trybu analizowania można znaleźć w temacie Konfigurowanie trybu analizowania obiektów biznesowych dla modułów i bibliotek.

Relacje

Relacja jest powiązaniem między co najmniej dwoma obiektami danych, zazwyczaj obiektami biznesowymi. W produkcie IBM Business Process Manager Advanced relacje można wykorzystywać do transformowania danych, które są równoważne w różnych obiektach biznesowych i w ramach różnych danych, ale są reprezentowane w odmienny sposób. Można ich również użyć do tworzenia powiązań między różnymi obiektami znajdującymi się w różnych aplikacjach. Relacje można współużytkować w aplikacjach, w różnych rozwiązaniach, a nawet w różnych produktach.

Usługa relacji w produkcie IBM Business Process Manager Advanced udostępnia infrastrukturę oraz operacje służące do zarządzania relacjami. Ponieważ umożliwia ona obsługę obiektów biznesowych niezależnie od tego, gdzie znajdują się te obiekty, może ona udostępniać całościowy, ujednoczony widok wszystkich aplikacji w przedsiębiorstwie i może służyć jako element budowania rozwiązań do zarządzania procesami biznesowymi (BPM). Ze względu na fakt, że relacje są rozszerzalne i mogą podlegać zarządzaniu, istnieje możliwość używania ich w złożonych rozwiązaniach integracji.

Czym są relacje?

Relacja jest powiązaniem między obiektami biznesowymi. Każdy obiekt biznesowy w relacji jest nazywany *uczestnikiem* relacji. Każdy uczestnik relacji jest wyróżniany od innych uczestników na podstawie funkcji lub *rol*, jaką pełni w tej relacji. Relacja zawiera listę ról.

Definicja relacji opisuje wszystkie role oraz powiązania między nimi. Opisuje również ogólny „kształt” relacji. Jedna rola może mieć na przykład tylko jednego uczestnika, ale inna rola może mieć dowolną liczbę uczestników. Użytkownik może na przykład zdefiniować relację *samochód-właściciel*, w której wiele samochodów może mieć jednego właściciela. Jedna instancja może mieć przykładowo następujących uczestników dla każdej z tych ról:

- Samochód (Ferrari)
- Właściciel (Jan)

Definicja relacji jest szablonem dla *instancji* relacji. Instancja jest obiektem czasu wykonywania dla relacji. W przykładzie *samochód-właściciel* instancja może opisywać dowolne spośród poniższych powiązań:

- Jan jest właścicielem samochodu Ferrari
- Anna jest właścicielem samochodu Mazda
- Tomasz jest właścicielem samochodu Ferrari

Dzięki relacjom użytkownik nie musi budować niestandardowych trwałych mechanizmów śledzenia relacji w ramach logiki biznesowej. W niektórych scenariuszach usługa relacji realizuje całość pracy za użytkownika. Patrzy przykład opisany w sekcji Relacje tożsamościowe.

Scenariusze

Poniżej przedstawiono typowy przykład sytuacji, w której rozwiązanie integracji może wykorzystywać relacje. Duża korporacja dokonuje zakupu wielu form lub jednostek organizacyjnych. Każda jednostka organizacyjna używa innego oprogramowania do monitorowania personelu i komputerów notebook. Firma wymaga sposobu monitorowania jej pracowników oraz ich komputerów notebook. Potrzebne jest rozwiązanie, które umożliwi realizowanie następujących działań:

- Wyświetlanie wszystkich pracowników w poszczególnych jednostkach organizacyjnych tak, jakby znajdowali się w jednej bazie danych.
- Wyświetlanie pojedynczego widoku wszystkich komputerów notebook używanych przez tych pracowników.
- Zezwolenie pracownikom na logowanie się w systemie i dokonanie zakupu komputera notebook.
- Obsługa różnych systemów aplikacji korporacyjnych w poszczególnych jednostkach organizacyjnych.

Aby to zrealizować, firma potrzebuje mechanizmu, który zagwarantuje że Jan Kowalski i Jan Krzysztof Kowalski w różnych aplikacjach są postrzegani jako ten sam pracownik. Niezbędny jest na przykład sposób konsolidowania pojedynczego obiektu w wielu obszarach aplikacji.

Bardziej złożone scenariusze relacji obejmują budowanie procesów BPEL, które uwzględniają relacje między różnymi obiektami w różnych aplikacjach. W złożonych scenariuszach relacji obiekty biznesowe znajdują się w rozwiązaniu integracji, a nie w aplikacjach. Usługa relacji udostępnia platformę do trwałego zarządzania relacjami. Przed wprowadzeniem usługi relacji użytkownik musiał budować własną usługę utrwalania obiektów. Poniżej przedstawiono dwa przykłady złożonych scenariuszy relacji:

- W aplikacji SAP istnieje obiekt biznesowy **samochód** z numerem nadwozia, a użytkownik chce śledzić, że ten samochód jest własnością określonej osoby. Jednak relacja własności dotyczy osoby, informacje o której są umieszczone w aplikacji PeopleSoft. W tym wzorcu relacji istnieją dwa rozwiązania i konieczne jest utworzenie między nimi mostu.
- Duża firma handlu detalicznego chce monitorować towary zwracane w celu uzyskania zwrotu gotówki lub kredytu. W tym procesie są uwzględniane dwie aplikacje: system zarządzania zamówieniami do obsługi zakupów oraz system obsługi zwrotów do obsługi zwrotów. Obiekty biznesowe znajdują się w więcej niż jednej aplikacji i konieczny jest sposób wyświetlania relacji między nimi.

Wspólne wzorce użycia

Najpopularniejszymi wzorcami relacji są wzorce *równoważności*. Są one oparte na odwołaniach lub na korelacji. Istnieją dwa typy relacji, które są zgodne z tym wzorcem: *nietożsamościowe* i *tożsamościowe*.

- **Relacje nietożsamościowe** ustanawiają powiązania między obiektami biznesowymi lub innymi danymi w konfiguracji jeden-do-wielu lub wiele-do-wielu. Dla każdej instancji relacji może istnieć jedna lub więcej instancji każdego uczestnika. Jednym z typów relacji nietożsamościowych jest statyczna relacja słownikowa. Jej przykładem może być relacja, w której kod CA w aplikacji SAP jest powiązany z kodem **Kalifornia** w aplikacji Siebel.

•

Relacje tożsamościowe ustanawiają powiązania między obiektami biznesowymi lub innymi danymi w konfiguracji jeden-do-jednego. Dla każdej instancji relacji może istnieć tylko jedna instancja każdego uczestnika. Relacje tożsamościowe przechwytyją odwołania między obiektami biznesowymi, które są semantycznie równoważne, ale są identyfikowane w różny sposób w poszczególnych aplikacjach. Każdy uczestnik relacji jest powiązany z obiektem biznesowym, który ma wartość (lub kombinację wartości) unikalnie identyfikującą obiekt. Relacje tożsamościowe zwykle transformują atrybuty kluczowe obiektów biznesowych, takie jak identyfikatory i kody produktów.

Jeśli na przykład w aplikacjach SAP, PeopleSoft i Siebel istnieje obiekt biznesowy **samochód** należy zbudować rozwiązanie, które synchronizuje je ze sobą, zazwyczaj konieczne byłoby wprowadzenie ręcznie utworzonej logiki synchronizacji relacji w postaci sześciu odwzorowań:

SAP -> ogólne
 ogólne -> SAP
 PeopleSoft-> ogólne
 ogólne -> PeopleSoft
 Siebel -> ogólne
 ogólne -> Siebel

Jeśli jednak w rozwiązaniu wykorzystywane są relacje, usługa relacji udostępnia wstępnie zbudowane implementacje wzorca, które utrzymują dla użytkownika wszystkie te instancje relacji.

Narzędzia do pracy z relacjami

Edytor relacji w produkcie Integration Designer jest narzędziem służącym do modelowania i projektowania relacji oraz ról integracji biznesowej. Szczegółowe informacje uzupełniające oraz informacje o czynnościach dotyczące tworzenia relacji i używania edytora relacji zawiera sekcja Tworzenie relacji.

Usługa relacji jest usługą infrastruktury w produkcie IBM Business Process Manager, która utrzymuje relacje i role w systemie oraz udostępnia operacje służące do zarządzania relacjami i rolami.

Menedżer relacji jest interfejsem administracyjnym służącym do zarządzania relacjami. Dostęp do niego można uzyskać za pośrednictwem stron menedżera relacji w Konsoli administracyjnej.

Relacje można wywoływać programowo za pośrednictwem funkcji API usługi relacji.

Usługa relacji

Usługa relacji zapisuje dane relacji w tabelach relacji, w których śledzi ona wartości specyficzne dla aplikacji w poszczególnych aplikacjach i w różnych rozwiązaniach. Usługa relacji udostępnia operacja do zarządzania relacjami i rolami.

Jak działają relacje

Relacje i role są definiowane przy użyciu graficznego interfejsu narzędzia edytora relacji w produkcie Integration Designer. Usługa relacji zapisuje dane korelacji w tabelach w bazie danych relacji w domyślnym źródle danych określanym podczas konfigurowania usługi relacji. W oddzielnej tabeli (nazywanej czasem tabelą uczestników) zapisywane są informacje na temat każdego uczestnika w relacji. Usługa relacji używa tych tabel relacji do śledzenia odpowiednich wartości specyficznych dla aplikacji oraz do propagowania zaktualizowanych informacji do wszystkich rozwiązań.

Relacje, które są artefaktami biznesowymi, są wdrażane w projekcie lub w bibliotece współużytkowanej. Podczas pierwszego wdrożenia usługa relacji zapełnia dane.

Instancje relacji są aktualizowane lub pobierane w czasie wykonywania, kiedy odwzorowania lub inne komponenty produktu IBM Business Process Manager wymagają instancji relacji. Jest to zależne od scenariusza.

Dane instancji relacji i roli mogą być manipulowane przy użyciu trzech sposobów:

- Wywołań fragmentów kodu Java funkcji API usługi relacji komponentu produktu IBM Business Process Manager.
- Transformacji relacji w usłudze odwzorowywania obiektów biznesowych produktu IBM Business Process Manager.
- Przy użyciu narzędzia menedżera relacji.

Szczegółowe informacje uzupełniające oraz informacje o czynnościach dotyczące tworzenia relacji, identyfikowania typów relacji oraz używania edytora relacji zawiera temat Tworzenie relacji.

Menedżer relacji

Menedżer relacji jest interfejsem administracyjnym służącym do zarządzania relacjami. Dostęp do niego można uzyskać za pośrednictwem stron menedżera relacji w Konsoli administracyjnej.

Menedżer relacji udostępnia graficzny interfejs użytkownika służący do tworzenia danych relacji i ról oraz manipulowania nimi w czasie wykonywania. Jednostkami relacji można zarządzać na wszystkich poziomach: instancji relacji, instancji roli oraz danych atrybutu i danych właściwości. Za pomocą menedżera relacji można realizować następujące działania:

- Wyświetlanie listy relacji w systemie oraz szczegółowych informacji na temat poszczególnych relacji.
- Zarządzanie instancjami relacji:
 - Realizowanie zapytań o dane relacji w celu wyświetlenia podzbiorów danych instancji.
 - Realizowanie zapytań o dane relacji w celu wyświetlenia podzbiorów danych instancji przy użyciu widoków bazy danych.
 - Wyświetlanie listy instancji relacji, które są zgodne z zapytaniem o relację, a także szczegółowych informacji na temat instancji.
 - Edytowanie wartości właściwości dla instancji relacji.
 - Tworzenie i usuwanie instancji relacji.
- Zarządzanie rolami i instancjami roli:
 - Wyświetlanie szczegółów roli lub instancji roli.
 - Edytowanie właściwości instancji roli.
 - Tworzenie i usuwanie instancji roli dla relacji.
 - Wycofywanie danych instancji relacji do punktu w czasie, w którym jest pewne, że są one poprawne.
- Importowanie danych z istniejącej relacji statycznej do systemu lub eksportowanie danych z istniejącej relacji statycznej do pliku RI lub CSV.

- Usuwanie schematów i danych relacji z repozytorium po zdeinstalowaniu aplikacji wykorzystującej te schematy i dane.

Relacje w środowiskach wdrożenia sieciowego

W środowiskach wdrożenia sieciowego można używać relacji bez konieczności dodatkowego konfigurowania.

W środowiskach wdrożenia sieciowego relacje są instalowane w klastrze aplikacji. Relacje są następnie widoczne w ramach klastra, a wszystkie serwery w klastrze mają dostęp do danych instancji przechowywanych w bazie danych relacji. Możliwość uruchamiania usługi relacji w środowisku wdrożenia sieciowego powoduje, że środowisko to charakteryzuje się skalowalnością i wysoką dostępnością.

Menedżer relacji umożliwia zarządzanie relacjami w różnych klastrach za pośrednictwem scentralizowanego interfejsu administracyjnego. Połączenie menedżera relacji z serwerem w klastrze jest nawiązywane poprzez wybór komponentu MBean relacji tego serwera.

Funkcje API usługi relacji

Relacje można wywoływać programowo przy użyciu funkcji API usługi relacji. Mogą być one wywoływane wewnątrz odwzorowań obiektów biznesowych lub poza nimi.

Dostępne są trzy typy funkcji API:

- Funkcje API do manipulowania instancjami relacji (między innymi do bezpośredniego tworzenia, aktualizowania i usuwania danych instancji).
- Funkcje API do obsługi wzorców relacji (między innymi correlate(), correlateforeignKeyLookup).
- Wzorce wyszukiwania relacji (funkcje API wyszukiwania).

Magistrala usług korporacyjnych w produkcie IBM Business Process Manager

Produkt IBM Business Process Manager obsługuje integrowanie usług aplikacji, obejmujące takie same możliwości jak produkt WebSphere Enterprise Service Bus.

Łączenie usług za pośrednictwem magistrali usług korporacyjnych

Magistrala usług korporacyjnych (ESB) umożliwia uzyskanie maksymalnej elastyczności w architekturze SOA. Uczestnicy interakcji usług łączą się z magistralą ESB zamiast bezpośrednio ze sobą.

Kiedy requester usług nawiązuje połączenie z magistralą ESB, magistrala ESB przejmuje odpowiedzialność za dostarczanie jego żądań do dostawcy usług za pomocą komunikatów, realizując w ten sposób wymaganą funkcję i zapewniając jakość usług. Magistrala ESB ułatwia interakcje między requesterami i dostawcami oraz rozwiązuje problemy związane z niezgodnymi protokołami, wzorcami interakcji lub możliwościami usług. Magistrala ESB może również umożliwiać lub usprawniać monitorowanie i zarządzanie. Magistrala ESB udostępnia opcje wirtualizacji i zarządzania, które implementują i rozszerzają podstawowe możliwości architektury SOA.

Magistrala ESB zapewnia obsługę następujących kwestii:

Położenie i tożsamość

Uczestnicy nie muszą znać położenia ani tożsamości innych uczestników. Requestery nie muszą na przykład wiedzieć, że żądanie może być obsłużone przez dowolnego spośród kilku dostawców, a dostawcy usług mogą być dodawani lub usuwani bez przerw w pracy.

Protokół interakcji

Uczestnicy nie muszą współużytkować tego samego protokołu komunikacyjnego ani stylu interakcji. Na przykład żądanie wyrażane w postaci komunikatu SOAP przez protokół HTTP może być obsłużone przez dostawcę, który umożliwia stosowanie tylko protokołu SOAP za pośrednictwem usługi JMS (Java Message Service).

Interfejs

Requestery i dostawcy nie muszą korzystać ze wspólnego interfejsu. Magistrala ESB eliminuje różnice, transformując komunikaty żądań i odpowiedzi na format oczekiwany przez dostawcę.

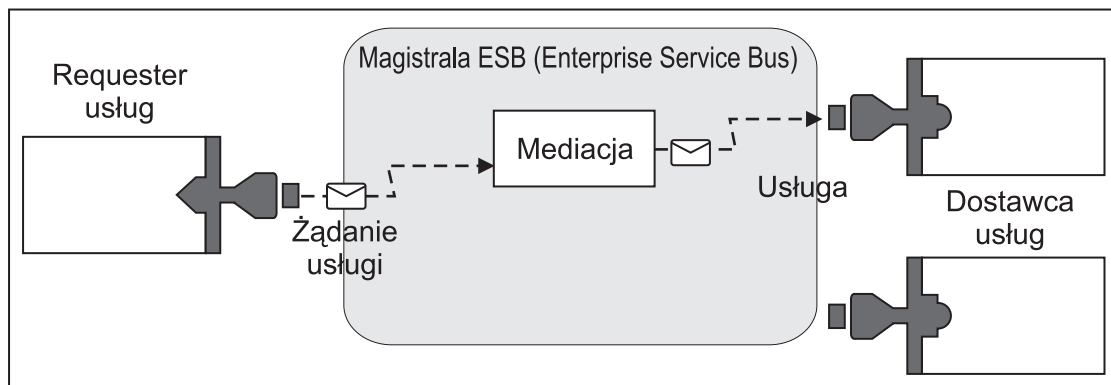
Jakość usług (interakcji)

Uczestnicy lub administratorzy systemu deklarują swoje oczekiwania w zakresie jakości usług, między innymi autoryzację żądań, szyfrowanie i deszyfrowanie treści komunikatów, automatyczną kontrolę interakcji usług oraz sposób kierowania żądań (na przykład optymalizacja pod kątem szybkości lub kosztu).

Umieszczenie magistrali ESB między uczestnikami pozwala modulować ich interakcję za pośrednictwem konstrukcji logicznej zwanej *mediacją*. Mediacje przeprowadzają operacje na wiadomościach w trakcie przetwarzania, kiedy są one przesyłane między requesterami i dostawcami. Mediacji można na przykład użyć do znalezienia usług o określonych parametrach zgłoszonych przez requester, a także do rozstrzygania różnic interfejsu między requesterami i dostawcami. W przypadku złożonych interakcji mediacje można łączyć sekwencyjnie w łańcuchy.

Przy użyciu mediacji magistrala usług korporacyjnych realizuje następujące działania między requesterem i usługą:

- *Kierowanie* komunikatów między usługami. Magistrala usług korporacyjnych oferuje wspólną infrastrukturę komunikacji, której można użyć do łączenia usług, a więc również funkcji biznesowych, jakie reprezentują. Nie wymaga to od programistów pisania i utrzymywania złożonej logiki połączeń.
- *Przekształcanie* protokołów transportu między requesterem i usługą. Magistrala usług korporacyjnych udostępnia spójny, oparty na standardach sposób integrowania funkcji biznesowych korzystających z różnych standardów informatycznych. Umożliwia to integrację funkcji biznesowych, które w innym przypadku nie mogłyby komunikować się ze sobą, na przykład łączenie aplikacji w silosach znajdujących się w działach lub uczestniczenie aplikacji z różnych firm w interakcji usług.
- *Transformowanie* formatów komunikatów między requesterem i usługą. Magistrala usług korporacyjnych umożliwia funkcjom biznesowym wymianę informacji w różnych formatach. Magistrala gwarantuje, że informacje dostarczane do funkcji biznesowej są w formacie wymaganym przez daną aplikację.
- *Obsługa* zdarzeń biznesowych z odmiennych źródeł. Oprócz wymiany komunikatów w celu obsługi żądań usług magistrala usług korporacyjnych obsługuje również interakcje oparte na zdarzeniach.



Rysunek 42. Magistrala usług korporacyjnych. Magistrala usług korporacyjnych kieruje komunikaty między aplikacjami, które są requesterami lub dostawcami usług. Magistrala przekształca protokoły transportowe i transformuje formaty komunikatów między requesterami i dostawcami. Na tym rysunku każda aplikacja używa innego protokołu (protokoły są reprezentowane przez różne kształty geometryczne konektorów) i używa różnych formatów komunikatów.

Jeśli używana jest magistrala usług korporacyjnych, użytkownik może skupić się na podstawowej działalności zamiast na systemach komputerowych. W razie potrzeby istnieje możliwość zmiany lub dodawania usług na przykład w reakcji na zmiany w wymaganiach biznesowych. Można również dodawać moc obliczeniową do obsługi usług lub nowe możliwości. Wymaganych zmian można dokonać, rekonfigurując magistralę bez wpływu na istniejące usługi i aplikacje wykorzystujące magistralę lub z niewielkim wpływem na nie.

Infrastruktura przesyłania komunikatów magistrali usług korporacyjnych

Produkt IBM Business Process Manager zawiera możliwości magistrali usług korporacyjnych. Produkt IBM Business Process Manager obsługuje integrację technologii, które są zorientowane na usługi oraz obsługę komunikatów, a także sterowane zdarzeniami. Taka integracja pozwala udostępniać opartą na standardach infrastrukturę przesyłania komunikatów w zintegrowanej magistrali usług korporacyjnych.

Możliwości usług korporacyjnych, które mogą być wykorzystane na potrzeby aplikacji korporacyjnych, udostępniają nie tylko warstwę transportową, ale również obsługę mediacji, która ułatwia interakcję między usługami. Magistrala usług korporacyjnych jest zbudowana na podstawie otwartych standardów oraz architektury zorientowanej na usługi (SOA). Jest ona oparta na stabilnej infrastrukturze Java EE oraz powiązanych usługach platformy udostępnianych przez wdrożenie sieciowe serwera IBM WebSphere Application Server.

Produkt IBM Business Process Manager jest oparty na tej samej technologii, która jest dostępna w produkcie IBM WebSphere Enterprise Service Bus. Ta możliwość stanowi część bazowej funkcjonalności produktu IBM Business Process Manager i do wykorzystania możliwości nie jest wymagana żadna dodatkowa licencja na produkt WebSphere Enterprise Service Bus.

Istnieje jednak możliwość wdrożenia w przedsiębiorstwie dodatkowych autonomicznych licencji produktu WebSphere Enterprise Service Bus, co pozwala rozszerzyć zasięg połączeń rozwiązań integracji procesów opartych na produkcie IBM Business Process Manager. Produkt WebSphere Enterprise Service Bus może być na przykład zainstalowany w miejscu bliższym aplikacji SAP, w celu udostępniania adaptera IBM WebSphere Adapter for SAP oraz w celu transformowania komunikatów SAP zanim informacje zostaną wysłane przez sieć do procesu biznesowego opartego na produkcie IBM Business Process Manager.

Hosty przesyłania komunikatów lub miejsca docelowego kolejki:

Host przesyłania komunikatów lub miejsca docelowego kolejki udostępnia funkcję przesyłania komunikatów na serwerze. Serwer staje się hostem miejsca docelowego przesyłania komunikatów, gdy zostanie skonfigurowany jako miejsce docelowe przesyłania komunikatów.

Mechanizm przesyłania komunikatów działa na serwerze. Mechanizm ten udostępnia funkcje przesyłania komunikatów oraz punkt pozwalający aplikacjom na nawiązywanie połączenia z magistralą. Kolejki komunikatów w mechanizmie przesyłania komunikatów są używane przez komunikację asynchroniczną architektury SCA (Service Component Architecture), importy oraz eksporty JMS, a także podczas asynchronicznego przetwarzania wewnętrznego.

Środowisko wdrażania tworzy podczas wdrażania modułów aplikacji połączenie między źródłem komunikatu i celem komunikatu przy użyciu magistrali. Wiedza dotycząca źródła komunikatu oraz celu komunikatu ułatwia określanie typu wymaganego środowiska wdrażania.

Aplikacje mogą przechowywać dane trwale w składnicy danych, która jest zestawem tabel w bazie danych lub schemacie, albo w składnicy plików. Mechanizm przesyłania komunikatów używa instancji źródła danych JDBC do interakcji z tą bazą danych.

Host docelowy przesyłania komunikatów należy skonfigurować podczas definiowania środowiska wdrażania przy użyciu opcji **Serwer** z poziomu Konsoli administracyjnej. Ewentualnie jako host docelowy należy wyznaczyć serwer podczas instalacji oprogramowania.

Składnice danych:

Każdy mechanizm przesyłania komunikatów może wykorzystywać składnicę danych będącą zbiorem tabel w bazie danych lub schematem, w którym zapisywane są dane trwałe.

Wszystkie tabele w składnicy danych są przechowywane w tym samym schemacie bazy danych. Istnieje możliwość utworzenia każdej składnicy danych w oddzielnej bazie danych. Można również utworzyć wiele składnic danych w tej samej bazie danych, w której każda składnica danych wykorzystuje inny schemat.

Mechanizm przesyłania komunikatów korzysta z instancji źródła danych JDBC w celu współdziałania z bazą danych zawierającą składnicę danych dla tego mechanizmu przesyłania komunikatów.

Dostawcy JDBC:

Dostawcy JDBC umożliwiają współdziałanie z aplikacjami korzystającymi z relacyjnych baz danych.

Aplikacje używają dostawców JDBC do pracy z relacyjnymi bazami danych. Dostawca JDBC udostępnia określoną klasę implementacji sterownika JDBC w celu uzyskania dostępu do określonego typu bazy danych. Aby utworzyć pulę połączeń do tej bazy danych, należy powiązać źródło danych z dostawcą JDBC. Dostawca JDBC i obiekty źródła danych pod względem funkcjonalnym odpowiadają fabryce połączeń JCA (Java EE Connector Architecture), która udostępnia połączenie z nierelacyjną bazą danych.

Należy zapoznać się z przykładami zarówno typowej konfiguracji środowiska autonomicznego, jak i typowej konfiguracji środowiska wdrażania przedstawionymi w poprzednim temacie.

Więcej informacji o dostawcach JDBC można znaleźć w temacie “Dostawcy JDBC” w Centrum informacyjnym produktu WebSphere Application Server.

Magistrala integracji usług dla produktu IBM Business Process Manager:

Magistrala integracji usług to zarządzany mechanizm komunikacji obsługujący integrację usług przy użyciu synchronicznego i asynchronicznego przesyłania komunikatów. W skład magistrali wchodzi połączone mechanizmy przesyłania komunikatów służące do zarządzania zasobami magistrali. Jest to jedna z technologii serwera WebSphere Application Server, na których oparto produkt IBM Business Process Manager.

Pojedyncza magistrala integracji usług i pojedynczy mechanizm przesyłania komunikatów domyślnie korzystają z tego samego schematu bazy danych jako bazy danych produktu. Każde środowisko wdrażania ma swoją własną magistralę. Pojedyncza magistrala ma nazwę **BPM.nazwa_środowiska_wdrażania.Bus**.

Miejsce docelowe magistrali to adres logiczny, do którego może zostać przyłączona aplikacja jako producent, konsument lub zarówno jako producent, jak i konsument. Miejsce docelowe kolejki to miejsce docelowe magistrali używane na potrzeby przesyłania komunikatów w trybie punkt z punktem.

Każda magistrala może zawierać jeden lub kilka elementów, z których każdy to serwer lub klaster.

Topologia magistrali jest fizycznym układem serwerów aplikacji, mechanizmów przesyłania komunikatów oraz menedżerów kolejek produktu WebSphere MQ oraz wzorcem połączeń magistrali i powiązań między nimi. Wspólnie tworzą one magistralę usług korporacyjnych.

Aplikacje usług i moduły usług

Moduł usług jest modułem SCA (Service Component Architecture), który udostępnia usługi w środowisku wykonawczym. Po wdrożeniu modułu usług w produkcie IBM Business Process Manager budowana jest powiązana aplikacja usług, która jest pakowana jako plik archiwum korporacyjnego (EAR).

Moduły usług są podstawowymi jednostkami wdrażania i mogą zawierać komponenty, biblioteki oraz moduły pomostowe używane przez powiązaną aplikację usług. Moduły usług mają eksporty oraz opcjonalnie importy, które definiują relacje między modułami oraz requesterami i dostawcami usług. Komponent WebSphere Process Server obsługuje moduły dla usług biznesowych i modułów mediacji. Zarówno moduły jak i moduły mediacji są typami modułów SCA. Moduł mediacji umożliwia komunikację między aplikacjami, wykonując transformację wywołania usługi do formatu zrozumiałego w miejscu docelowym, przekazując żądanie do miejsca docelowego i zwracając wynik do nadawcy. Moduł usługi biznesowej implementuje logikę procesu biznesowego. Jednak moduł może zawierać tę samą logikę mediacji, którą można umieścić w module mediacji.

Wdrażanie aplikacji usług

Proces wdrażania pliku EAR zawierającego aplikację usług jest taki sam jak proces wdrażania dowolnego pliku EAR. Istnieje możliwość modyfikowania wartości dla parametrów mediacji w czasie wdrażania. Po wdrożeniu pliku EAR zawierającego moduł SCA można wyświetlić szczegóły aplikacji usług oraz powiązanego z nią modułu. Użytkownik może zobaczyć, w jaki sposób moduł usług jest połączony z requesterami usług (za pośrednictwem eksportów) oraz dostawcami usług (za pośrednictwem importów).

Wyświetlanie szczegółów modułu SCA

Dostępne do wyświetlenia szczegóły modułu usług są zależne od modułu SCA. Zawierają one następujące atrybuty:

- nazwa modułu SCA,
- opis modułu SCA,
- nazwę powiązanej aplikacji,
- informacje o wersji modułu SCA, jeśli moduł podlega kontroli wersji,
- importy modułu SCA:
 - Interfejsy importu są abstrakcyjnymi definicjami opisującymi sposób uzyskiwania dostępu do usługi przez moduł SCA.
 - Powiązania importu są konkretnymi definicjami określającymi mechanizm fizyczny używany przez moduł SCA w celu uzyskania dostępu do usługi. Przykład: SOAP/HTTP.
- eksporty modułu SCA:
 - Interfejsy eksportu są abstrakcyjnymi definicjami, które opisują sposób uzyskiwania dostępu do modułu SCA przez requestery usług.
 - Powiązania eksportu są konkretnymi definicjami określającymi mechanizm fizyczny używany przez requester usług w celu uzyskania dostępu do modułu SCA oraz pośrednio do usługi.
- właściwości modułu SCA.

Importy i powiązania importu:

Importy definiują interakcje między modułami SCA a dostawcami usług. Moduły SCA wykorzystują importy, aby umożliwić komponentom uzyskiwanie dostępu do usług zewnętrznych (usług znajdujących się poza modułem SCA) przy użyciu reprezentacji lokalnej. Powiązania importu definiują konkretny sposób, w jaki uzyskiwany jest dostęp do usługi zewnętrznej.

Jeśli moduły SCA nie wymagają dostępu do usług zewnętrznych, nie muszą mieć importów. Moduły mediacji zazwyczaj mają co najmniej jeden import, który służy do przekazywania komunikatów lub żądań do miejsc docelowych.

Interfejsy i powiązania

Import modułu SCA wymaga co najmniej jednego interfejsu i ma jedno powiązanie.

- Interfejsy importu są abstrakcyjnymi definicjami określającymi zbiór operacji przy użyciu języka WSDL (Web Services Description Language). Jest to język XML służący do opisywania usług Web Services. Moduł SCA może mieć wiele interfejsów importu.
- Powiązania importu są konkretnymi definicjami określającymi mechanizm fizyczny używany przez moduły SCA w celu uzyskania dostępu do usługi zewnętrznej.

Obsługiwane powiązania importu

Produkt IBM Business Process Manager obsługuje następujące powiązania importu:

- Powiązania SCA łączą moduły SCA z innymi modułami SCA. Powiązania SCA są zwane również powiązaniami domyślnymi.

- Powiązania usług Web Services umożliwiają komponentom wywoływanie usług Web Services. Obsługiwane są protokoły SOAP1.1/HTTP, SOAP1.2/HTTP i SOAP1.1/JMS.
Istnieje możliwość użycia powiązania SOAP1.1/HTTP lub SOAP1.2/HTTP na podstawie interfejsu JAX-WS (Java API for XML Web Services). Umożliwia to interakcję z usługami wykorzystującymi powiązania dokumentu lub powiązania literału RPC i wykorzystuje procedury obsługi JAX-WS do dostosowywania wywołań. Dostępne jest oddzielne powiązanie SOAP1.1/HTTP, które umożliwia interakcję z usługami używającymi powiązań kodowanych RPC, oraz które może być stosowane w sytuacji, kiedy wymagane jest używanie procedur obsługi JAX-RPC do dostosowywania wywołań.
- Powiązania HTTP umożliwiają dostęp do aplikacji za pośrednictwem protokołu HTTP.
- Powiązania importu EJB (Enterprise JavaBeans) umożliwiają komponentom SCA wywoływanie usług udostępnianych przez logikę biznesową Java EE działającą na serwerze Java EE.
- Powiązania EIS (Enterprise Information System) udostępniają połączenia między komponentami SCA i zewnętrznym systemem EIS. Ta komunikacja jest realizowana za pośrednictwem adapterów zasobów.
- Powiązania JMS (Java Message Service) 1.1 umożliwiają współdziałanie z domyślnym dostawcą przesyłania komunikatów serwera WebSphere Application Server. Powiązania JMS mogą wykorzystywać różne typy transportu, między innymi TCP/IP, HTTP lub HTTPS. Obsługiwana jest automatycznie klasa komunikatu JMS oraz jej pięć podtypów (tekst, bajty, obiekt, strumień i odwzorowanie).
- Ogólne powiązania JMS umożliwiają współdziałanie z dostawcami JMS innych firm, które są zintegrowane z serwerem WebSphere Application Server przy użyciu narzędzia serwera aplikacji JMS (ASF).
- Powiązania WebSphere MQ JMS umożliwiają współdziałanie z dostawcami JMS opartymi na produkcie WebSphere MQ. Obsługiwana jest automatycznie klasa komunikatu JMS oraz jej pięć podtypów (tekst, bajty, obiekt, strumień i odwzorowanie). Aby używać produktu WebSphere MQ jako dostawcy JMS, należy skorzystać z powiązań JMS WebSphere MQ.
- Powiązania WebSphere MQ umożliwiają współdziałanie z produktem WebSphere MQ. Powiązań WebSphere MQ można używać tylko ze zdalnymi menedżerami kolejek za pośrednictwem połączenia klienta produktu WebSphere MQ. Nie można używać ich z lokalnymi menedżerami kolejek. Aby komunikować się z rodzimymi aplikacjami WebSphere MQ, należy użyć powiązań WebSphere MQ.

Dynamiczne wywoływanie usług

Usługi można wywoływać za pośrednictwem dowolnego obsługiwane powiązania importu. Usługa zazwyczaj znajduje się w punkcie końcowym określonym w imporcie. Ten punkt końcowy jest nazywany statycznym punktem końcowym. Istnieje możliwość wywołania innej usługi, poprzez nadpisanie statycznego punktu końcowego. Dynamiczne nadpisywanie statycznych punktów końcowych pozwala wywołać usługę w innym punkcie końcowym za pośrednictwem dowolnego obsługiwane powiązania importu. Dynamiczne wywoływanie usług pozwala również wywołać usługę w sytuacji, gdy obsługiwane powiązanie importu nie ma statycznego punktu końcowego.

Do określenia protokołu i jego konfiguracji na potrzeby wywołania dynamicznego używany jest import z powiązaniem. Import używany na potrzeby wywołania dynamicznego może być połączony z komponentem wywołującym lub może być wybierany dynamicznie w czasie wykonywania.

W przypadku wywołań usług Web Services i wywołań SCA można również wykonać wywołanie dynamiczne bez importu, na podstawie protokołu i konfiguracji pobranej z adresu URL punktu końcowego. Typ miejsca docelowego wywołania jest identyfikowany na podstawie adresu URL punktu końcowego. Jeśli używany jest import, adres URL musi być zgodny z protokołem powiązania importu.

- Adres URL architektury SCA oznacza wywołanie innego modułu SCA.
- Adres URL protokołu HTTP lub usługi JMS domyślnie oznacza wywołanie usługi Web Service. W przypadku tych adresów URL można podać dodatkową wartość typu powiązania, która określa, czy adres URL reprezentuje wywołanie za pośrednictwem powiązania HTTP czy powiązania JMS.
- W przypadku adresu URL protokołu URL dla usługi Web Service domyślną opcją jest użycie protokołu SOAP 1.1, ale można określić wartość typu powiązania, która wskazuje że należy użyć protokołu SOAP 1.2.

Eksporty i powiązania eksportów:

Eksporty definiują interakcje między modułami SCA a requesterami usług. Moduły SCA wykorzystują eksporty w celu udostępniania usług. Powiązania eksportu definiują konkretny sposób, w jaki requestery usług uzyskują dostęp do modułu SCA.

Interfejsy i powiązania

Eksport modułu SCA wymaga co najmniej jednego interfejsu.

- Interfejsy eksportu są abstrakcyjnymi definicjami określającymi zbiór operacji przy użyciu języka WSDL (Web Services Description Language). Jest to język XML służący do opisywania usług Web Services. Moduł SCA może mieć wiele interfejsów eksportu.
- Powiązania eksportu są konkretnymi definicjami określającymi mechanizm fizyczny używany przez requestery usług w celu uzyskania dostępu do usługi. Eksport modułu SCA ma zazwyczaj zdefiniowane jedno powiązanie. Eksport bez określonego powiązania jest interpretowany przez środowisko wykonawcze jako eksport z powiązaniem SCA.

Obsługiwane powiązania eksportu

Produkt IBM Business Process Manager obsługuje następujące powiązania eksportu:

- Powiązania SCA łączą moduły SCA z innymi modułami SCA. Powiązania SCA są zwane również powiązaniami domyślnymi.
- Powiązanie usługi Web Service umożliwia wywoływanie eksportów jako usług Web Services. Obsługiwane są protokoły SOAP1.1/HTTP, SOAP1.2/HTTP i SOAP1.1/JMS.

Istnieje możliwość użycia powiązania SOAP1.1/HTTP lub SOAP1.2/HTTP na podstawie interfejsu JAX-WS (Java API for XML Web Services). Umożliwia to interakcję z usługami wykorzystującymi powiązania dokumentu lub powiązania literału RPC i wykorzystuje procedury obsługi JAX-WS do dostosowywania wywołań. Dostępne jest oddzielne powiązanie SOAP1.1/HTTP, które umożliwia interakcję z usługami używającymi powiązań kodowanych RPC, oraz które może być stosowane w sytuacji, kiedy wymagane jest używanie procedur obsługi JAX-RPC do dostosowywania wywołań.

- Powiązania HTTP umożliwiają dostęp do eksportów za pośrednictwem protokołu HTTP.
- Powiązania EJB (Enterprise JavaBeans) umożliwiają ujawnianie komponentów SCA jako komponentów EJB, co pozwala na wywoływanie komponentów SCA przez logikę biznesową Java EE (komponenty te nie są dostępne dla tej logiki w innej sytuacji).
- Powiązania EIS (Enterprise Information System) udostępniają połączenia między komponentami SCA i zewnętrznym systemem EIS. Ta komunikacja jest realizowana za pośrednictwem adapterów zasobów.
- Powiązania JMS (Java Message Service) 1.1 umożliwiają współdziałanie z domyślnym dostawcą przesyłania komunikatów serwera WebSphere Application Server. Powiązania JMS mogą wykorzystywać różne typy transportu, między innymi TCP/IP, HTTP lub HTTPS. Obsługiwana jest automatycznie klasa komunikatu JMS oraz jej pięć podtypów (tekst, bajty, obiekt, strumień i odwzorowanie).
- Ogólne powiązania JMS umożliwiają współdziałanie z dostawcami JMS innych firm, które są zintegrowane z serwerem WebSphere Application Server przy użyciu narzędzia serwera aplikacji JMS (ASF).
- Powiązania WebSphere MQ JMS umożliwiają współdziałanie z dostawcami JMS opartymi na produkcie WebSphere MQ. Obsługiwana jest automatycznie klasa komunikatu JMS oraz jej pięć podtypów (tekst, bajty, obiekt, strumień i odwzorowanie). Aby używać produktu WebSphere MQ jako dostawcy JMS, należy skorzystać z powiązań JMS WebSphere MQ.
- Powiązania WebSphere MQ umożliwiają współdziałanie z produktem WebSphere MQ. Do nawiązania połączenia z menedżerem kolejek MQ na zdalnym komputerze wykorzystywane jest połączenie zdalne (połączenie klienta). Połączenie lokalne (połączenie powiązania) jest połączeniem bezpośrednim z produktem WebSphere MQ. Może ono być używane tylko do nawiązania połączenia z menedżerem kolejek MQ na tym samym komputerze. Produkt WebSphere MQ pozwala stosować oba typy połączeń, ale powiązania MQ obsługują tylko połączenie zdalne (połączenie klienta).

Moduły mediacji:

Moduły mediacji są to moduły SCA (Service Component Architecture), które umożliwiają zmianę formatu, treści oraz miejsca docelowego żądań usług.

Moduły mediacji działają na komunikatach, które są w trakcie przetwarzania między requesterami usług i dostawcami usług. Użytkownik może kierować komunikaty do różnych dostawców usług oraz zmieniać treść lub format komunikatu. Moduły mediacji mogą udostępniać funkcje, takie jak rejestrowanie komunikatów czy przetwarzanie błędów, dostosowane do konkretnych wymagań.

Niektóre aspekty modułów mediacji można zmieniać z Konsoli administracyjnej bez konieczności ponownego wdrażania modułu.

Komponenty modułów mediacji

Moduły mediacji zawierają następujące elementy:

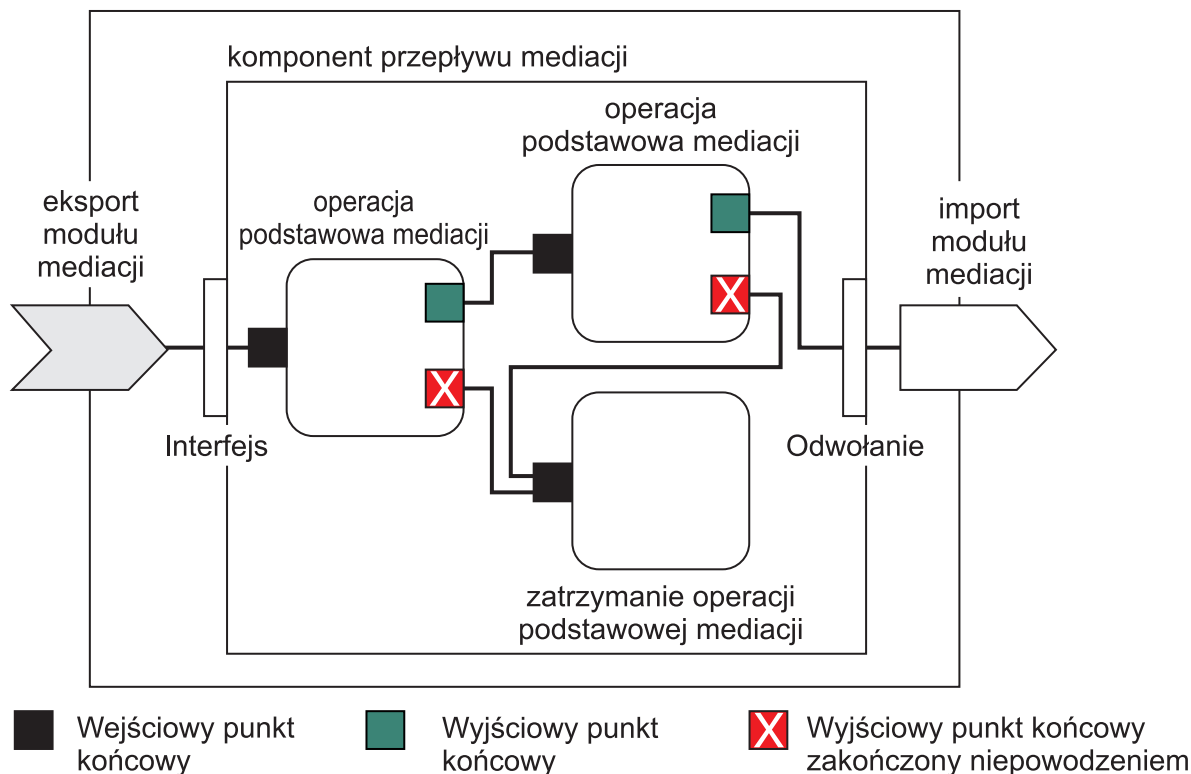
- Importy, które definiują transakcje między modułami SCA i dostawcami usług. Umożliwiają one modułom SCA wywoływanie usług zewnętrznych w taki sposób, jakby były one usługami lokalnymi. Użytkownik może wyświetlać importy modułów mediacji oraz modyfikować powiązanie.
- Eksporty, które definiują interakcje między modułami SCA a requesterami usług. Umożliwiają one modułom SCA oferowanie usług i definiowanie interfejsów zewnętrznych (punktów dostępu) modułu SCA. Użytkownik może wyświetlać eksporty modułu mediacji.
- Komponenty SCA, które stanowią elementy składowe modułów SCA, na przykład modułów mediacji. Istnieje możliwość graficznego tworzenia i dostosowywania modułów SCA przy użyciu produktu Integration Designer. Po wdrożeniu modułu mediacji można dostosować niektóre aspekty tego modułu z poziomu Konsoli administracyjnej bez konieczności ponownego wdrażania modułu.

Moduły mediacji zazwyczaj zawierają konkretny typ komponentu SCA nazywany *komponentem przepływu mediacji*. Komponenty przepływu mediacji definiują przepływy mediacji.

Komponent przepływu mediacji może zawierać dowolną liczbę operacji podstawowych mediacji (żadnych operacji, jedną operację lub określoną liczbę operacji). IBM Business Process Manager obsługuje dostarczany zestaw operacji podstawowych mediacji, które udostępniają funkcje do kierowania i transformowania komunikatów. Aby zwiększyć elastyczność w zakresie operacji podstawowej mediacji, można użyć niestandardowej operacji podstawowej mediacji w celu wywołania niestandardowej logiki.

Celem modułu mediacji, który nie zawiera komponentu przepływu mediacji, jest transformowanie żądań usług z jednego protokołu na inny. Można na przykład utworzyć żądanie usługi przy użyciu protokołu SOAP/JMS i przed przesłaniem dalej przekształcić je do formatu SOAP/HTTP.

Uwaga: Z poziomu produktu IBM Business Process Manager można wyświetlać moduły mediacji i dokonywać w nich pewnych zmian. Z poziomu produktu IBM Business Process Manager nie można jednak wyświetlać ani zmieniać komponentów SCA znajdujących się w module. Do dostosowywania komponentów SCA służy produkt Integration Designer.



Rysunek 43. Uproszczony przykład modułu mediacji. Moduł mediacji zawiera jeden komponent przepływu mediacji, który zawiera operacje podstawowe mediacji.

- Właściwości

Operacje podstawowe mediacji mają właściwości. Niektóre z tych właściwości mogą być wyświetlane w Konsoli administracyjnej jako dodatkowe właściwości modułu SCA.

Aby właściwości operacji podstawowej mediacji były widoczne z poziomu Konsoli administracyjnej produktu IBM Business Process Manager, programista integracji musi awansować właściwości. Niektóre właściwości umożliwiają konfigurowanie przez administratora, a produkt Integration Designer opisuje te właściwości jako właściwości możliwe do awansowania, ponieważ mogą być one awansowane z cyklu integracji do cyklu administracyjnego. Inne właściwości nie nadają się do konfigurowania przez administratora, ponieważ ich modyfikowanie może wpłynąć na przepływ mediacji w taki sposób, że będzie konieczne ponowne wdrożenie modułu mediacji. Program Integration Designer wyświetla właściwości, które można zmienić, aby wypromować właściwości operacji podstawowej mediacji.

Istnieje możliwość użycia Konsoli administracyjnej produktu IBM Business Process Manager do zmiany wartości awansowanych właściwości bez konieczności ponownego wdrażania modułu mediacji ani restartowania serwera lub modułu.

Najczęściej przepływy mediacji korzystają natychmiast ze zmian właściwości. Jeśli jednak zmiany właściwości występują w komórce menedżera wdrażania, są one uwzględniane w każdym węźle po synchronizacji tego węzła. Przepływy mediacji, które są w trakcie przetwarzania, nadal używają wcześniejszych wartości.

Uwaga: Z poziomu Konsoli administracyjnej można zmienić tylko wartości właściwości, ale nie można zmienić grup właściwości, ich nazw ani typów. Aby zmienić grupy właściwości, ich nazwy lub typy, należy użyć produktu Integration Designer.

- Moduł mediacji lub zależna biblioteka mogą również definiować przepływy podrzędne. Przepływ podrzędny hermetyzuje zbiór operacji podstawowych mediacji połączonych ze sobą jako element logiki integracji wielokrotnego użytku. W celu wywołania przepływu podrzędnego do przepływu mediacji można dodać operację podstawową.

Wdrażanie modułów mediacji

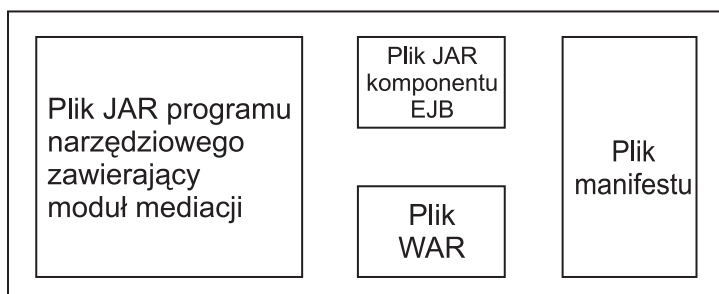
Moduły mediacji są tworzone przy użyciu produktu Integration Designer i są zazwyczaj wdrażane w produkcie IBM Business Process Manager wewnątrz pliku archiwum korporacyjnego (EAR).

Wartość awansowanych właściwości można zmieniać w czasie wdrażania.

Moduł mediacji można wyeksportować z produktu Integration Designer, co powoduje, że produkt Integration Designer spakuje moduł mediacji wewnątrz pliku archiwum Java (JAR), a plik JAR wewnątrz pliku EAR. Plik EAR można następnie wdrożyć, instalując nową aplikację z poziomu Konsoli administracyjnej.

Moduły mediacji mogą być postrzegane jako pojedynczy obiekt. Moduły SCA są jednak definiowane przez kilka plików XML zapisanych w pliku JAR.

Przykład pliku EAR zawierającego moduł mediacji



Rysunek 44. Uproszczony przykład pliku EAR zawierającego moduł mediacji. Plik EAR zawiera pliki JAR. Plik programu narzędziowego JAR zawiera moduł mediacji.

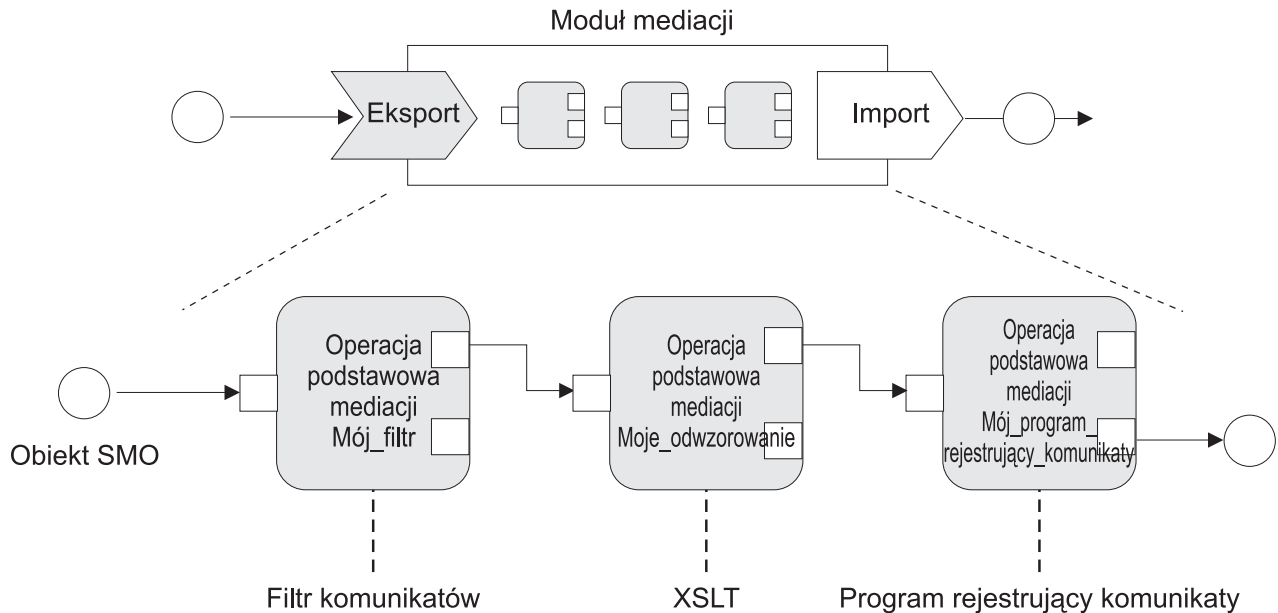
Operacje podstawowe mediacji:

Komponenty przepływu mediacji opierają się na przepływach komunikatów między komponentami usług. Możliwości komponentu mediacji są implementowane przez *operacje podstawowe mediacji*, które implementują standardowe typy implementacji usług.

Komponent przepływu mediacji ma co najmniej jeden przepływ. Na przykład: jeden przepływ dla żądania i jeden dla odpowiedzi.

IBM Business Process Manager obsługuje dostarczany zbiór operacji podstawowych mediacji, które implementują standardowe możliwości mediacji dla modułów mediacji lub modułów wdrożonych w produkcie IBM Business Process Manager. Jeśli potrzebne są specjalne możliwości mediacji, użytkownik może utworzyć własne niestandardowe operacje podstawowe mediacji.

Operacja podstawowa mediacji definiuje operację “wejścia”, która przetwarza lub obsługuje komunikaty reprezentowane przez obiekty komunikatów usługi (Service Message Object - SMO). Operacja podstawowa mediacji może również definiować operacje “wyjścia”, które wysyłają komunikaty do innego komponentu lub modułu.



Rysunek 45. Moduł mediacji zawierający trzy operacje podstawowe mediacji

Produktu Integration Designer można użyć do konfigurowania operacji podstawowych mediacji i ustawiania ich właściwości. Niektóre z tych właściwości mogą być uwidaczniane administratorowi środowiska wykonawczego poprzez ich awansowanie. Dowolna właściwość operacji podstawowej mediacji, która może być awansowana, może być również właściwością dynamiczną. Właściwość dynamiczna może być nadpisana w czasie wykonywania przy użyciu pliku strategii.

Integration Designer umożliwia również graficzne modelowanie i składanie komponentów przepływu mediacji z operacji podstawowych mediacji oraz składanie modułów mediacji lub modułów na podstawie komponentów przepływu mediacji. Konsola administracyjna odnosi się do modułów mediacji i modułów jako modułów SCA.

Integration Designer umożliwia również definiowanie przepływów podrzędnych w modułach lub zależnych od nich bibliotekach. Przepływ podrzędny może zawierać dowolną operację podstawową mediacji oprócz operacji podstawowej mediacji Rozstrzygnięcie strategii. Przepływ podrzędny jest wywoływany z poziomu przepływu żądania lub odpowiedzi lub z poziomu innego przepływu podrzędnego przy użyciu operacji podstawowej mediacji Przepływ podrzędny. Właściwości awansowane z operacji podstawowych mediacji w przepływie podrzędnym są ujawniane jako właściwości w operacjach podstawowych mediacji Przepływ podrzędny. Mogą być następnie ponownie awansowane, aż do poziomu modułu, gdzie mogą być modyfikowane przez administratora środowiska wykonawczego.

Obsługiwane operacje podstawowe mediacji

IBM Business Process Manager obsługuje następujący zbiór operacji podstawowych mediacji:

Odzworowanie obiektu biznesowego

Transformuje komunikaty.

- Definiuje transformacje komunikatów przy użyciu odzworowania obiektu biznesowego, które może być ponownie wykorzystywane.
- Umożliwia graficzne definiowanie transformacji komunikatów przy użyciu edytora odzworowań obiektów biznesowych.
- Może zmieniać treść komunikatu.
- Może transformować typ komunikatu wejściowego na inny typ komunikatu wyjściowego.

Mediacja niestandardowa

Umożliwia zaimplementowanie własnej logiki mediacji w kodzie Java. Operacja podstawowa mediacji

Mediacja niestandardowa łączy elastyczność operacji podstawowej mediacji zdefiniowanej przez użytkownika z prostotą predefiniowanej operacji podstawowej mediacji. Istnieje możliwość tworzenia złożonych transformacji i wzorców kierowania poprzez:

- Tworzenie kodu Java.
- Tworzenie własnych właściwości.
- Dodawanie nowych punktów końcowych.

Z poziomu operacji podstawowej mediacji Mediacja niestandardowa można wywoływać usługę, ale do wywoływania usług zaprojektowana jest operacja podstawowa mediacji Wywołanie usługi, która udostępnia dodatkowe funkcje takie jak ponawianie prób.

Procedura obsługi danych

Umożliwia transformowanie części komunikatu. Służy do przekształcenia elementu komunikatu z formatu fizycznego w strukturę logiczną lub odwrotnie. Podstawowym zastosowaniem tej operacji podstawowej mediacji jest przekształcanie formatu fizycznego, na przykład łańcucha Tekst w obiekcie komunikatu tekstu JMS, w logiczną strukturę obiektu biznesowego i odwrotnie. Ta mediacja jest często używana do realizowania następujących działań:

- Transformowania sekcji komunikatu wejściowego ze zdefiniowanej struktury na inną. Przykład: jeśli obiekt SMO zawiera wartość łańcucha rozdzielaną przecinkami, a użytkownik chce przeanalizować tę wartość i umieścić dane w konkretnym obiekcie biznesowym.
- Zmiany typu komunikatu. Przykład: jeśli eksport JMS został skonfigurowany tak, aby używał powiązania danych podstawowego typu JMS, a programista integracji zdecyduje, że treść wewnątrz modułu mediacji powinna zostać przekształcona w konkretną strukturę obiektu biznesowego.

Wyszukiwanie w bazie danych

Modyfikuje komunikaty przy użyciu informacji z bazy danych dostarczonej przez użytkownika.

- Konieczne jest skonfigurowanie bazy danych, źródła danych oraz wszystkich ustawień uwierzytelniania na serwerze, które mają być używane przez operację podstawową mediacji Wyszukiwanie w bazie danych. W tym celu należy użyć Konsoli administracyjnej.
- Operacja podstawowa mediacji Wyszukiwanie w bazie danych może odczytywać dane z tylko jednej tabeli.
- Określona kolumna klucza musi zawierać wartości unikalne.
- Dane w kolumnach wartości muszą być danymi typu prostego schematu XML lub typu schematu XML, który rozszerza typ prostego schematu XML.

Wyszukiwanie punktów końcowych

Umożliwia dynamiczne kierowanie żądań poprzez wyszukiwanie punktów końcowych usług w repozytorium.

- Informacje o punktach końcowych usług są pobierane z produktu WebSphere Service Registry and Repository (WSRR). Rejestr WSRR może być rejestrem lokalnym lub zdalnym.
- Zmiany w rejestrze są dokonywane z poziomu Konsoli administracyjnej produktu WSRR.
- IBM Business Process Manager musi wiedzieć, jaki rejestr ma być wykorzystywany, a więc należy utworzyć definicje dostępu do rejestru WSRR przy użyciu Konsoli administracyjnej produktu IBM Business Process Manager.

Emiter zdarzeń

Rozszerza monitorowanie poprzez możliwość wysyłania zdarzeń z wewnątrz komponentu przepływu mediacji.

- Działanie mediowania można zawiesić, anulując zaznaczenie pola wyboru.
- Zdarzenia emitera zdarzeń można wyświetlać przy użyciu przeglądarki modelu CBE (Common Base Events) w produkcie IBM Business Process Manager.
- Zdarzenia należy wysyłać jedynie w znaczącym punkcie przepływu mediacji (z powodów związanych z wydajnością).
- Istnieje możliwość zdefiniowania części komunikatu zawartego w zdarzeniu.
- Zdarzenia są wysyłane w postaci zdarzeń modelu CBE (Common Base Events) i są wysyłane do serwera infrastruktury CEI (Common Event Infrastructure).

- Aby w pełni wykorzystać informacje Emitera zdarzeń, konsumenci zdarzeń muszą rozumieć strukturę zdarzeń CBE. Zdarzenia CBE mają ogólny schemat, ale nie jest to model danych konkretnej aplikacji, które są zawarte w elementach danych rozszerzonych. Aby modelować elementy danych rozszerzonych, narzędzia produktu Integration Designer generują plik definicji katalogu zdarzeń infrastruktury CEI (Common Event Infrastructure) dla każdej skonfigurowanej operacji podstawowej mediacji Emiter zdarzeń. Pliki definicji katalogu zdarzeń są artefaktami eksportu, które są udostępniane w celu pomocy użytkownikowi. Nie są one używane przez produkt Integration Designer ani przez środowisko wykonawcze produktu IBM Business Process Manager. Użytkownik powinien odwoływać się do plików definicji katalogu zdarzeń podczas tworzenia aplikacji w celu konsumowania zdarzeń Emitera zdarzeń.
- Istnieje możliwość określenia innego monitorowania z poziomu produktu IBM Business Process Manager. Można na przykład monitorować zdarzenia, które mają być emitowane z importów i eksportów.

Niepowodzenie

Zatrzymuje konkretną ścieżkę z przepływie i generuje wyjątek.

Zwielokrotnianie wejściowe

Pomaga agregować (składać) komunikaty.

- Może być używana tylko w połączeniu z operacją podstawową mediacji Zwielokrotnianie wyjściowe.
- Operacje podstawowe mediacji Zwielokrotnianie wejściowe i Zwielokrotnianie wyjściowe wspólnie umożliwiają agregowanie danych w pojedynczym komunikacie wyjściowym.
- Operacja podstawowa mediacji Zwielokrotnianie wejściowe odbiera wiadomości, aż zostanie osiągnięty punkt decyzyjny, a następnie tworzony jest jeden komunikat wyjściowy.
- Do przechowywania danych agregacji należy użyć kontekstu współużytkowanego.

Zwielokrotnianie wyjściowe

Pomaga dzielić i agregować (składać) komunikaty.

- Operacje podstawowe mediacji Zwielokrotnianie wejściowe i Zwielokrotnianie wyjściowe wspólnie umożliwiają agregowanie danych w pojedynczym komunikacie wyjściowym.
- W trybie iteracji operacja podstawowa mediacji Zwielokrotnianie wyjściowe umożliwia iterację pojedynczego komunikatu wejściowego zawierającego powtarzający się element. Dla każdego wystąpienia powtarzającego się elementu wysyłany jest komunikat.
- Do przechowywania danych agregacji należy użyć kontekstu współużytkowanego.

Procedura ustawiająca nagłówki HTTP

Udostępnia mechanizm do zarządzania nagłówkami w komunikatach HTTP.

- Umożliwia tworzenie, ustawianie, kopiowanie lub usuwanie nagłówków komunikatu HTTP.
- Umożliwia ustawienie wielu działań w celu zmiany wielu nagłówków HTTP.

Odwzorowanie

Transformuje komunikaty.

- Umożliwia wykonywanie transformacji XSL (Extensible Stylesheet Language) lub transformacji odwzorowania obiektu biznesowego.
- Komunikaty są transformowane przy użyciu transformacji XSLT 1.0 lub XSLT 2.0 bądź transformacji odwzorowania obiektu biznesowego. Transformacje XSL działają na serializowanym komunikacie XML, natomiast transformacje odwzorowania obiektu biznesowego działają na obiektach Service Data Object (SDO).

Procedura ustawiająca elementy komunikatów

Udostępnia prosty mechanizm ustawiania treści komunikatów.

- Umożliwia zmianę, dodawanie lub usuwanie elementów komunikatu.
- Nie zmienia typu komunikatu.
- Dane w kolumnach wartości muszą być danymi typu prostego schematu XML lub typu schematu XML, który rozszerza typ prostego schematu XML.

Filtr komunikatów

Kieruje komunikaty do różnych ścieżek, na podstawie treści komunikatu.

- Działanie mediowania można zawiesić, anulując zaznaczenie pola wyboru.

Program rejestrujący komunikaty

Rejestruje komunikaty w relacyjnej bazie danych lub za pośrednictwem niestandardowego programu rejestrującego użytkownika. Komunikaty są zapisywane w formacie XML, a więc dane mogą być później przetwarzane przez aplikacje obsługujące format XML.

- Działanie mediowania można zawiesić, anulując zaznaczenie pola wyboru.
- Schemat relacyjnej bazy danych (struktura tabel) jest zdefiniowany przez IBM.
- Domyślnie operacja podstawowa mediacji Program rejestrujący komunikaty wykorzystuje wspólną bazę danych. Środowisko wykonawcze odwzorowuje źródło danych **jdbc/mediation/messageLog** na wspólną bazę danych.
- Istnieje możliwość ustawienia klas implementacji procedury obsługi, aby dostosować działanie niestandardowego programu rejestrującego. Można opcjonalnie udostępnić klasy implementacji formatera, klasy implementacji filtru lub obie te klasy, aby dostosować działanie niestandardowego programu rejestrującego.

Procedura ustawiająca nagłówek MQ

Udostępnia mechanizm do zarządzania nagłówkami w komunikatach MQ.

- Umożliwia tworzenie, ustawianie, kopiowanie lub usuwanie nagłówków komunikatu MQ.
- Umożliwia ustawienie wielu działań w celu zmiany wielu nagłówków MQ.

Rozstrzygnięcie strategii

Umożliwia dynamiczne konfigurowanie żądań poprzez wyszukiwanie punktów końcowych usług i powiązanych plików strategii w repozytorium.

- Istnieje możliwość użycia pliku strategii do dynamicznego nadpisywania awansowanych właściwości innych operacji podstawowych mediacji.
- Informacje o punktach końcowych usług i informacje o strategiach są pobierane z produktu WebSphere Service Registry and Repository (WSRR). Rejestr WSRR może być rejestrem lokalnym lub zdalnym.
- Zmiany w rejestrze są dokonywane z poziomu Konsoli administracyjnej produktu WSRR.
- IBM Business Process Manager musi wiedzieć, jaki rejestr ma być wykorzystywany, a więc należy utworzyć definicje dostępu do rejestru WSRR przy użyciu Konsoli administracyjnej produktu IBM Business Process Manager.

Wywołanie usługi

Wywołuje usługę z wewnątrz przepływu mediacji, zamiast czekać na koniec przepływu mediacji i używać mechanizmu wywołania.

- Jeśli usługa zwróci błąd, można ponownie wywołać tę samą usługę lub wywołać inną.
- Operacja podstawowa mediacji Wywołanie usługi jest zaawansowaną operacją podstawową mediacji, której można używać samodzielnie na potrzeby prostych wywołań usług lub w połączeniu z innymi operacjami podstawowymi mediacji w celu realizacji złożonych mediacji.

Ustawianie typu komunikatu

Podczas programowania integracji umożliwia traktowanie pól komunikatów o słabych typach tak, jakby miały stały typ. Pole ma słaby typ, jeśli może zawierać więcej niż jeden typ danych. Pole ma stały typ, jeśli jego typ oraz struktura wewnętrzna są znane.

- W czasie wykonywania operacja podstawowa mediacji Ustawianie typu komunikatu umożliwia sprawdzenie, czy treść komunikatu jest zgodna z oczekiwanymi typami danych.

Procedura ustawiająca nagłówek SOAP

Udostępnia mechanizm do zarządzania nagłówkami w komunikatach SOAP.

- Umożliwia tworzenie, ustawianie, kopiowanie lub usuwanie nagłówków komunikatu SOAP.
- Umożliwia ustawienie wielu działań w celu zmiany wielu nagłówków SOAP.

Zatrzymywanie

Zatrzymuje konkretną ścieżkę w przepływie bez generowania wyjątku.

Filtr typów

Umożliwia kierowanie komunikatów do różnych ścieżek przepływu na podstawie ich typu.

Pobieranie z produktu WebSphere eXtreme Scale

Istnieje możliwość pobierania informacji ze środowiska pamięci podręcznej serwera eXtreme Scale.

- Można wyszukać wartości w pamięci podręcznej i zapisać je jako elementy w komunikacie przy użyciu klucza.
- Łącząc operacje podstawowe mediacji Zapisywanie i Pobieranie z produktu eXtreme Scale, można umieścić w pamięci podręcznej odpowiedź z systemu zaplecza. Późniejsze żądania nie będą musiały uzyskiwać dostępu do tego systemu zaplecza.
- Należy utworzyć definicje produktu eXtreme Scale przy użyciu Konsoli administracyjnej produktu WebSphere ESB, aby umożliwić określenie serwera eXtreme Scale, który powinien być używany.

Zapisywanie w produkcie WebSphere eXtreme Scale

Istnieje możliwość zapisania informacji w środowisku pamięci podręcznej serwera eXtreme Scale.

- Informacje można zapisać w pamięci podręcznej produktu eXtreme Scale przy użyciu klucza i obiektu.
- Łącząc operacje podstawowe mediacji Zapisywanie i Pobieranie z produktu eXtreme Scale, można użyć operacji podstawowej mediacji Zapisywanie do zapisania danych w obrębie pamięci podręcznej, a operacji podstawowej mediacji Pobieranie - do pobrania danych zapisanych wcześniej w pamięci podręcznej.
- Należy utworzyć definicje produktu eXtreme Scale przy użyciu Konsoli administracyjnej produktu WebSphere ESB, aby umożliwić określenie serwera eXtreme Scale, który powinien być używany.

Dynamiczne kierowanie:

Komunikaty można kierować na różne sposoby przy użyciu punktów końcowych zdefiniowanych w czasie integracji lub punktów końcowych określanych dynamicznie w czasie wykonywania.

Kierowanie dynamiczne obejmuje dwa następujące przypadki:

- Kierowanie komunikatów w sytuacji, w której przepływ jest dynamiczny, ale wszystkie możliwe punkty końcowe są predefiniowane w module SCA (Service Component Architecture).
- Kierowanie komunikatów w sytuacji, w której przepływ jest dynamiczny i wybór punktów końcowych jest również dynamiczny. Punkty końcowe usługi są wybierane ze źródła zewnętrznego w czasie wykonywania.

Dynamiczny wybór punktu końcowego

Środowisko wykonawcze ma możliwość kierowania komunikatów żądań i odpowiedzi do adresu punktu końcowego identyfikowanego przez element nagłówka komunikatu. Ten element nagłówka komunikatu może być aktualizowany przez operacje podstawowe mediacji w przepływie mediacji. Adres punktu końcowego może być aktualizowany przy użyciu informacji z rejestru, bazy danych lub przy użyciu informacji z samego komunikatu. Kierowanie komunikatów odpowiedzi ma zastosowanie tylko, jeśli odpowiedź jest wysyłana przez eksport JAX-WS usługi Web Service.

Aby środowisko wykonawcze implementowało kierowanie dynamiczne dla żądania lub odpowiedzi, moduł SCA musi mieć ustawioną właściwość **Używaj dynamicznego punktu końcowego**, jeśli jest ustawiony w nagłówku komunikatu. Programiści integracji mogą ustawić właściwość **Używaj dynamicznego punktu końcowego**, jeśli jest ustawiony w nagłówku komunikatu lub mogą awansować tę właściwość (uwidocznic ją w czasie wykonywania), aby mógł ją ustawić administrator środowiska wykonawczego. Właściwości modułu można wyświetlić w oknie **Właściwości modułu**. Aby wyświetlić to okno, należy wybrać opcję **Aplikacje > Moduły SCA > Właściwości modułu**. Programista integracji nadaje awansowanym właściwościom nazwy aliasów i te właśnie nazwy są wyświetlane w Konsoli administracyjnej.

Rejestr

Do przechowywania informacji o punktach końcowych usług można użyć produktu IBM WebSphere Service Registry and Repository (WSRR). Następnie można utworzyć moduły SCA do pobierania punktów końcowych z rejestru WSRR.

Podczas tworzenia modułów SCA użytkownik wykorzystuje operację podstawową mediacji Wyszukiwanie punktów końcowych, która umożliwia przepływowi mediacji wysłanie zapytania do rejestru WSRR w celu uzyskania punktu końcowego usługi lub zbioru punktów końcowych usługi. Jeśli moduł SCA pobiera zbiór punktów końcowych, musi użyć innej operacji podstawowej mediacji w celu wybrania preferowanego punktu końcowego.

Sterowanie żądaniami usług przy użyciu strategii mediacji:

Strategii mediacji można używać do sterowania przepływami mediacji między requesterami a dostawcami usług.

Przepływami mediacji można sterować przy użyciu strategii mediacji zapisanych w produkcie IBM WebSphere Service Registry and Repository (WSRR). Implementacja zarządzania strategiami usług w produkcie WSRR jest oparta na środowisku strategii usług Web Services (Web Services Policy Framework - WS-Policy).

Aby sterować żądaniami usług przy użyciu strategii mediacji w rejestrze produktu WSRR muszą znajdować się odpowiednie moduły SCA (Service Component Architecture) i dokumenty strategii mediacji.

Sposób przyłączania strategii mediacji do żądania usługi

Podczas tworzenia modułu SCA, który wymaga użycia strategii mediacji, do przepływu mediacji należy dołączyć operację podstawową mediacji Rozstrzygnięcie strategii. Operacja podstawowa mediacji Rozstrzygnięcie strategii uzyskuje informacje o strategii mediacji z rejestru w czasie wykonywania. Dlatego też moduł SCA musi zawierać komponent przepływu mediacji, aby obsługiwać sterowanie żądaniami usług przy użyciu strategii mediacji.

Istnieje możliwość przyłączenia co najmniej jednej strategii mediacji w rejestrze do modułu SCA lub do usługi docelowej używanej przez moduł SCA. Przyłączone strategie mediacji mogą być używane na potrzeby wszystkich komunikatów usług przetwarzanych przez ten moduł SCA (znajdują się w zasięgu). Strategie mediacji mogą zawierać załączniki strategii, które definiują warunki. Warunki strategii mediacji umożliwiają stosowanie różnych strategii mediacji w różnych kontekstach. Strategie mediacji mogą dodatkowo zawierać klasyfikacje, których można użyć do określenia stanu nadzoru.

WebSphere Service Registry and Repository:

Produkt WebSphere Service Registry and Repository (WSRR) umożliwia zapisywanie informacji, uzyskiwanie dostępu do informacji oraz zarządzanie informacjami dotyczącymi punktów końcowych usług oraz strategii mediacji. Produktu WSRR można użyć, aby aplikacje usług były bardziej dynamiczne i łatwiejsze do dostosowania do zmiennych warunków biznesowych.

Wprowadzenie

Przepływy mediacji mogą używać produktu WSRR jako mechanizmu dynamicznego wyszukiwania, udostępniającego informacje o punktach końcowych usług lub strategiach mediacji.

Aby skonfigurować dostęp do produktu WSRR, należy utworzyć dokumenty definicji WSRR przy użyciu Konsoli administracyjnej. Można również użyć komend administrowania produktem WSRR z klienta skryptowego wsadmin. Definicje WSRR oraz właściwości ich połączeń są mechanizmem używanym w celu nawiązywania połączenia z instancją rejestru i pobierania punktu końcowego usługi lub strategii mediacji.

Punkty końcowe usług

Produktu WSRR można użyć do zapisywania informacji o używanych usługach, usługach, których użycie jest planowane lub usługach, o których użytkownik chce mieć informacje. Te usługi mogą znajdować się w systemach użytkownika lub w innych systemach. Aplikacja może na przykład użyć produktu WSRR do znalezienia najbardziej odpowiedniej usługi w celu spełnienia potrzeb funkcjonalnych i związanych z wydajnością.

Podczas tworzenia modułu SCA, który wymaga dostępu do punktów końcowych usług z produktu WSRR, do przepływu mediacji należy dołączyć operację podstawową mediacji Wyszukiwanie punktów końcowych. Operacja

podstawowa mediacji Wyszukiwanie punktów końcowych uzyskuje punkty końcowe usług z rejestru w czasie wykonywania.

Strategie mediacji

Produktu WSRR można również użyć do zapisywania informacji o strategii mediacji. Strategie mediacji pomagają sterować żadaniami usług, dynamicznie nadpisując właściwości modułu. Jeśli produkt WSRR zawiera strategię mediacji, które są przyłączone do obiektu reprezentującego moduł SCA lub usługę docelową, strategię mediacji mogą nadpisywać właściwości modułu. Aby stosować różne strategię mediacji w różnych kontekstach, można utworzyć warunki strategii mediacji.

Uwaga: Strategie mediacji dotyczą sterowania przepływem mediacji, a nie zabezpieczeń.

Podczas tworzenia modułu SCA, który wymaga użycia strategii mediacji, do przepływu mediacji należy dołączyć operację podstawową mediacji Rozstrzygnięcie strategii. Operacja podstawowa mediacji Rozstrzygnięcie strategii uzyskuje informacje o strategii mediacji z rejestru w czasie wykonywania.

WebSphere eXtreme Scale:

WebSphere eXtreme Scale (eXtreme Scale) umożliwia udostępnianie systemu zapisywania w pamięci podręcznej, który można zintegrować z produktem IBM Business Process Manager. Użycie produktu eXtreme Scale z produktem IBM Business Process Manager pozwala skrócić czasy odpowiedzi i zwiększyć niezawodność, a także udostępnić dodatkowe funkcje integracji.

eXtreme Scale działa jak elastyczna i skalowalna siatka danych w pamięci. Ta siatka danych dynamicznie buforuje, partycjonuje i replikuje dane aplikacji oraz logikę biznesową na wielu serwerach, a także pozwala nimi zarządzać. eXtreme Scale ułatwia także uzyskanie takich właściwości usługi, jak integralność transakcyjna, wysoka dostępność i przewidywalne czasy odpowiedzi.

Dostęp do funkcji buforowania produktu eXtreme Scale można uzyskać, umieszczając w obrębie przepływu operacje podstawowe mediacji produktu WebSphere eXtreme Scale. Podczas tworzenia modułu architektury Service Component Architecture (SCA), który wymaga zapisania informacji w pamięci podręcznej produktu eXtreme Scale należy dołączyć do przepływu mediacji operację podstawową mediacji Zapisywanie do produktu WebSphere eXtreme Scale. Aby pobrać informacje z pamięci podręcznej produktu eXtreme Scale, należy dołączyć operację podstawową mediacji Pobieranie z produktu WebSphere eXtreme Scale. Połączenie tych dwóch operacji podstawowych mediacji w przepływie mediacji umożliwia buforowanie odpowiedzi z systemu zaplecza, aby późniejsze żądania mogły ją pobrać z pamięci podręcznej.

Aby skonfigurować dostęp do produktu eXtreme Scale, należy utworzyć definicję produktu WebSphere eXtreme Scale przy użyciu Konsoli administracyjnej. Można również użyć komend administrowania produktem WebSphere eXtreme Scale z klienta skryptowego wsadmin. Definicja produktu eXtreme Scale jest mechanizmem używanym przez operacje podstawowe mediacji Zapisywanie i Pobieranie produktu WebSphere eXtreme Scale do łączenia się z serwerem eXtreme Scale.

Klienci usługi komunikatów

Dostępne są klienci usługi komunikatów dla środowisk C/C++ i .NET, które umożliwiają aplikacjom w językach innych niż Java nawiązywanie połączenia z magistralą usług korporacyjnych.

Message Service Clients for C/C++ and .NET udostępniają funkcję API o nazwie XMS, która ma ten sam zbiór interfejsów co interfejs API usługi JMS (Java Message Service). Klient usługi komunikatów dla środowiska C/C++ zawiera dwie implementacje funkcji API XMS - jedną do używania w aplikacjach w języku C i jedną do używania w aplikacjach w języku C++. Message Service Client for .NET zawiera w pełni zarządzaną implementację funkcji API XMS, która może być używana przez dowolny język zgodny ze środowiskiem .NET.

Klienci usługi komunikatów dla środowiska .NET można uzyskać pod adresem http://www-01.ibm.com/support/docview.wss?rs=0&q1=IA9H&uid=swg24011756&loc=en_US&cs=utf-8&cc=us&lang=en.

Klienty usługi komunikatów dla środowiska C/C++ można uzyskać pod adresem http://www-01.ibm.com/support/docview.wss?rs=0&q1=ia94&uid=swg24007092&loc=en_US&cs=utf-8&cc=us&lang=en.

Istnieje również możliwość zainstalowania i wykorzystywania obsługi klienta Java EE z produktu WebSphere Application Server Network Deployment. Zawiera on klienta usług Web Services, klienta EJB i klienta JMS.

Rozdział 2. Dodatkowe informacje o kluczowych pojęciach

Ta sekcja jest punktem wyjścia do zdobywania wiedzy o technologiach stosowanych w produkcie IBM Business Process Manager i wykorzystywanych przez ten produkt.

Scenariusze tworzenia treści

Scenariusze pozwalają zapoznać się z komponentami i produktami z dziedziny zarządzania procesami biznesowymi oraz rozpocząć z nimi pracę.

Kontrola wersji

Cykl życia aplikacji procesu rozpoczyna się od utworzenia aplikacji procesu i jest kontynuowany poprzez cykl aktualizowania, wdrażania, jednoczesnego wdrażania, wycofywania wdrożenia oraz archiwizowania aplikacji procesu. *Kontrola wersji* jest mechanizmem używanym do zarządzania cyklem życia aplikacji procesu poprzez jednoznaczne identyfikowanie poszczególnych wersji aplikacji procesu.

Sposób działania kontroli wersji w produkcie IBM Business Process Manager zależy od tego, jaki element jest wdrażany - czy jest to aplikacja procesu wdrażana z repozytorium komponentu IBM Process Center, czy aplikacja korporacyjna wdrażana bezpośrednio z poziomu produktu IBM Integration Designer.

Aplikacje procesów i biblioteki narzędziowe wdrażane w środowisku wykonawczym z komponentu Process Center są domyślnie objęte kontrolą wersji. W przypadku aplikacji korporacyjnych można wybrać opcję kontroli wersji modułów i bibliotek w produkcie IBM Integration Designer.

Istnieje dodatkowo możliwość tworzenia wersji czynności personelu lub automatu skończonego tak, aby w środowisku wykonawczym mogło jednocześnie istnieć wiele wersji czynności lub automatu skończonego.

Kontrola wersji aplikacji procesów

Kontrola wersji umożliwia środowisku wykonawczemu identyfikowanie obrazów stanu w cyklu życia aplikacji procesów oraz współbieżne uruchamianie wielu obrazów stanu na serwerze Process Server.

Aby zrozumieć, w jaki sposób działa kontrola wersji aplikacji procesów, należy pamiętać, że aplikacja procesu jest kontenerem przechowującym różne aplikacje używane przez aplikację procesu (na przykład modele procesów lub definicje BPD, odwołania do bibliotek narzędziowych, usługi lub ścieżki). Kontrola wersji jest realizowana na poziomie tego kontenera, a nie na poziomie pojedynczych artefaktów. W przypadku aplikacji procesów oznacza to, że kontrola wersji jest realizowana w momencie wykonania obrazu stanu.

Użytkownicy mogą porównywać obrazy stanu, aby określić różnice między wersjami. Jeśli na przykład pewien programista rozwiązał problem występujący w usłudze i wykonał w tym punkcie obraz stanu zawierającej ją aplikacji procesów lub biblioteki narzędziowej, a następnie inny programista wprowadził kilka dodatkowych zmian do tej samej usługi i wykonał nowy obraz stanu, to menedżer projektu może porównać te dwa obrazy stanu w celu określenia momentów wprowadzenia zmian oraz ich autorów. Jeśli menedżer projektu uzna, że dodatkowe zmiany wprowadzone do usługi były bez znaczenia, może on przywrócić projekt do stanu z momentu wykonania obrazu stanu zawierającego pierwotną poprawkę.

Istnieje możliwość uruchamiania różnych wersji (obrazów stanu) aplikacji procesu współbieżnie na serwerze. Podczas instalowania nowego obrazu stanu można usunąć oryginalny lub pozostawić go, aby nadal działał.

Kontekst wersji

Każdy obraz stanu ma unikalne metadane umożliwiające identyfikację wersji (zwane kontekstem wersji). Może to być ten identyfikator, ale zalecamy stosowanie trzysegmentowego systemu wersji liczbowych w formacie <wersja główna>.<wersja podrzędna>.<usługa>. Szczegółowy opis tego schematu kontroli wersji zawierają tematy dotyczące konwencji nazewnictwa.

Program IBM Business Process Manager przypisuje każdej aplikacji procesów globalną przestrzeń nazw. Globalna przestrzeń nazw jest albo najnowszą wersją aplikacji procesu, albo konkretnym obrazem stanu aplikacji procesu. Nazwa wersji używana przez serwer może się składać z maksymalnie siedmiu znaków, dlatego przypisana nazwa jest akronimem złożonym ze znaków z nazwy przypisanego obrazu stanu. Akronimy obrazów stanu są takie same, jak nazwy obrazów stanu, jeśli nazwy obrazów stanu są tworzone zgodnie z zaleceniami IBM i nie składają się z więcej niż siedmiu znaków. Na przykład nazwa obrazu stanu 1.0.0 będzie miała akronim 1.0.0, a nazwa obrazu stanu 10.3.0 będzie miała akronim 10.3.0. Zapewni to, że akronim obrazu stanu będzie unikalny w kontekście aplikacji procesów w zasięgu serwera Process Center. Dlatego nie można edytować akronimu obrazu stanu.

Uwagi dotyczące kontroli wersji aplikacji procesów w wielu klastrach

Tę samą wersję aplikacji procesów można zainstalować w wielu klastrach w tej samej komórce. Aby można było rozróżnić wiele instalacji tej samej wersji aplikacji procesów, dla każdej instalacji należy utworzyć obraz stanu i dodać do jej nazwy identyfikator unikalny w całej komórce (na przykład v1.0_kom1_1 i v1.0_kom1_2). Każdy obraz stanu jest nową wersją aplikacji procesów (z perspektywy zarządzania cyklem życia), jednak jego zawartość i funkcje są takie same.

Podczas instalowania aplikacji procesów w klastrze jest wykonywana automatyczna synchronizacja węzłów.

Uwagi dotyczące kontroli wersji bibliotek narzędziowych komponentu Process Designer

Należy pamiętać, że obrazy stanu aplikacji procesów są zazwyczaj wykonywane, kiedy użytkownik jest gotowy do testowania lub instalacji. Z kolei obraz stanu biblioteki narzędziowej jest tworzony, gdy jest ona gotowa do użycia przez aplikacje procesów. Aby w późniejszym czasie zaktualizować bibliotekę narzędziową, należy wykonać kolejny obraz stanu gotowej „najnowszej wersji”, a wtedy właściciele aplikacji procesów i bibliotek narzędziowych będą mogli zdecydować, czy chcą korzystać z tego nowego obrazu stanu.

Kontrola wersji modułów i bibliotek

Jeśli moduł lub biblioteka znajdują się w aplikacji procesu lub bibliotece narzędziowej, przyjmują one cykl życia aplikacji procesu lub biblioteki narzędziowej (wersje, obrazy stanu, ścieżki itp.). Nazwy modułów i bibliotek muszą być unikalne w zasięgu aplikacji procesu lub biblioteki narzędziowej.

Niniejszy temat opisuje kontrolę wersji modułów i bibliotek używanych z aplikacjami procesów. Należy jednak zauważyć, że jeśli moduły są wdrażane bezpośrednio z produktu IBM Integration Designer na serwerze Process Server, można nadal postępować zgodnie z procedurą przypisywania numerów wersji do modułów podczas wdrażania, która została opisana w sekcji “Tworzenie modułów i bibliotek z kontrolą wersji”.

Moduł lub biblioteka, które są powiązane z komponentem IBM Process Center, muszą mieć niezależne biblioteki w tej samej aplikacji procesu lub w zależnej bibliotece narzędziowej.

Poniższa tabela przedstawia wybory, jakich można dokonać w edytorze zależności produktu IBM Integration Designer, jeśli biblioteka jest powiązana z aplikacją procesu lub biblioteką narzędziową:

Tabela 36. Zależności dla modułu, aplikacji procesu lub biblioteki narzędziowej oraz bibliotek globalnych

Zasięg biblioteki	Opis	Możliwa zależność od . .
Moduł	Kopia tej biblioteki istnieje na serwerze dla każdego modułu korzystającego z tej biblioteki.	Biblioteka w zasięgu modułu może być zależna od wszystkich typów bibliotek.

Tabela 36. Zależności dla modułu, aplikacji procesu lub biblioteki narzędziowej oraz bibliotek globalnych (kontynuacja)

Zasięg biblioteki	Opis	Możliwa zależność od. . .
Aplikacja procesu lub biblioteka narzędziowa	Biblioteka jest współużytkowana przez wszystkie moduły w zasięgu aplikacji procesu lub biblioteki narzędziowej. To ustawienie obowiązuje, jeśli wdrożenie jest realizowane za pośrednictwem komponentu IBM Process Center. Jeśli wdrożenie jest realizowane poza komponentem IBM Process Center, biblioteka jest kopiowana do każdego modułu. Uwaga: Biblioteki utworzone w produkcie IBM Integration Designer 8 mają domyślnie poziom współużytkowania aplikacji procesu lub biblioteki narzędziowej .	Biblioteka tego typu może być zależna tylko od bibliotek globalnych.
Globalne	Biblioteka jest współużytkowana przez wszystkie działające moduły.	Biblioteka globalna może być zależna tylko od innych bibliotek globalnych. Uwaga: W celu wdrożenia biblioteki globalnej należy skonfigurować bibliotekę współużytkowaną produktu WebSphere. Więcej informacji na ten temat zawiera sekcja “Zależności modułów i bibliotek”.

Moduły i biblioteki powiązane z aplikacjami procesów lub bibliotekami narzędziowymi

Nie ma potrzeby kontroli wersji modułów ani bibliotek powiązanych z aplikacjami procesów lub bibliotekami narzędziowymi.

Moduły i biblioteki powiązane z aplikacją procesu lub biblioteką narzędziową nie wymagają kontroli wersji. W rzeczywistości nie można nawet utworzyć wersji modułu lub biblioteki, które są powiązane z aplikacją procesu lub biblioteką narzędziową w edytorze zależności. Moduły i biblioteki powiązane z aplikacją procesu lub biblioteką narzędziową korzystają z obrazów stanu, które są funkcją w komponencie Process Center. Ta funkcja pozwala uzyskać te same wyniki co kontrola wersji.

Biblioteki powiązane z aplikacją procesu lub biblioteką narzędziową nie mają wymaganego numeru wersji w sekcji Biblioteki edytora zależności, ponieważ nie jest potrzebna kontrola wersji.

Konwencje nazewnictwa

Konwencja nazewnictwa służy do rozróżnienia poszczególnych wersji aplikacji procesu w ramach cyklu życia obejmującego aktualizowanie, wdrażanie, jednoczesne wdrażanie, wycofywanie wdrożenia oraz archiwizowanie.

Niniejsza sekcja opisuje konwencje używane do jednoznacznego identyfikowania wersji aplikacji procesu.

Kontekst wersji jest kombinacją akronimów, która jednoznacznie opisuje aplikację procesu lub bibliotekę narzędziową. Każdy typ akronimu ma konwencję nazewnictwa. Akronim może mieć maksymalnie siedem znaków z zestawu znaków [A-Z0-9_], a akronim obrazu stanu może dodatkowo zawierać kropkę.

- Akronim aplikacji procesu jest tworzony podczas tworzenia aplikacji procesu. Może mieć on maksymalnie siedem znaków długości.
- Akronim obrazu stanu jest tworzony automatycznie podczas tworzenia obrazu stanu. Może mieć on maksymalnie siedem znaków długości.

Jeśli nazwa obrazu stanu spełnia kryteria poprawnego akronimu obrazu stanu, nazwa obrazu stanu i jego akronim będą takie same.

Uwaga: Jeśli używana jest funkcja kierowania z uwzględnieniem wersji komponentu przepływu mediacji, obraz stanu należy nazwać tak, aby był zgodny ze schematem <wersja>.<wydanie>.<modyfikacja> (na przykład **1.0.0**). Ponieważ akronim obrazu stanu może mieć maksymalnie siedem znaków, wartości cyfrowe mogą mieć maksymalnie łącznie pięć cyfr (pięć cyfr plus dwie kropki). Dlatego też należy uważać podczas zwiększania wartości pól cyfr, ponieważ wartość jest obcinana do pierwszych siedmiu znaków.

Na przykład akronimem dla nazwy obrazu stanu **11.22.33** będzie akronim **11.22.3**.

- Akronim ścieżki jest generowany automatycznie na podstawie pierwszego znaku każdego słowa w nazwie ścieżki. Jeśli na przykład zostanie utworzona nowa ścieżka o nazwie **Moja nowa sciezka**, wartością akronimu będzie **Mns**. Domyślną nazwą ścieżki i akronimem ścieżki jest **Main**. Akronim ścieżki jest uwzględniany w kontekście kontroli wersji podczas wdrożenia na serwerze komponentu IBM Process Center, jeśli ten akronim jest inny niż **Main** (Główny).

Definicja procesu biznesowego w aplikacji procesu jest zazwyczaj identyfikowana na podstawie akronimu nazwy aplikacji procesu, akronimu obrazu stanu oraz nazwy definicji procesu biznesowego. O ile to możliwe, należy wybierać unikalne nazwy definicji procesów biznesowych. Jeśli istnieją duplikaty nazw, mogą wystąpić następujące problemy:

- Ujawnianie definicji procesów biznesowych w postaci usług Web Services może być niemożliwe, chyba że zostanie zastosowany pewien rodzaj mediacji.
- Wywołanie definicji procesu biznesowego utworzonej w produkcie IBM Process Designer z poziomu procesu BPEL utworzonego w produkcie IBM Integration Designer może być niemożliwe.

Kontekst wersji jest różny, w zależności od sposobu wdrożenia aplikacji procesu.

Konwencje nazewnictwa dla wdrożeń serwera komponentu Process Center

Na serwerze komponentu IBM Process Center można wdrożyć obraz stanu aplikacji procesu, a także obraz stanu biblioteki narzędziowej. Można również wdrożyć najnowszą wersję aplikacji procesu lub najnowszą wersję biblioteki narzędziowej (*najnowsza wersja* to bieżąca wersja robocza aplikacji procesu lub biblioteki narzędziowej). Kontekst wersji jest różny, w zależności od typu wdrożenia.

W przypadku aplikacji procesów do jednoznacznego identyfikowania wersji używane są najnowszą wersją aplikacji procesu lub konkretny obraz stanu aplikacji procesu.

Biblioteki narzędziowe można wdrożyć z jedną aplikacją procesu lub kilkoma aplikacjami procesów, ale cykl życia każdej biblioteki narzędziowej jest powiązany z cyklem życia aplikacji procesu. Dla każdej aplikacji procesu na serwerze wdrożona jest jej własna kopia zależnej biblioteki narzędziowej lub zależnych bibliotek narzędziowych. Wdrożona biblioteka narzędziowa nie jest współużytkowana przez aplikacje procesów.

Jeśli ścieżka powiązana z aplikacją procesu ma inną nazwę niż nazwa domyślna **Main** (Główna), akronim ścieżki również stanowi część kontekstu wersji.

Więcej informacji na ten temat zawiera sekcja “Przykłady” na stronie 33 w dalszej części tego tematu.

Obrazy stanu aplikacji procesu

W przypadku wdrażania obrazów stanu aplikacji procesu kontekst wersji jest kombinacją następujących elementów:

- akronimu nazwy aplikacji procesu,
- akronimu ścieżki aplikacji procesu (jeśli używana jest ścieżka inna niż **Main**),
- Akronimu obrazu stanu aplikacji procesu

Autonomiczne biblioteki narzędziowe

W przypadku wdrażania obrazów stanu biblioteki narzędziowej kontekst wersji jest kombinacją następujących elementów:

- akronimu nazwy biblioteki narzędziowej,

- akronimu ścieżki biblioteki narzędziowej (jeśli używana jest ścieżka inna niż **Main**),
- akronimu obrazu stanu biblioteki narzędziowej.

Najnowsze wersje

Najnowsze wersje aplikacji procesu są stosowane podczas testowania iteracyjnego w produkcie Process Designer. Można je wdrażać tylko na serwerach komponentu Process Center.

W przypadku wdrażania najnowszych wersji aplikacji procesu kontekst wersji jest kombinacją następujących elementów:

- akronimu nazwy aplikacji procesu,
- akronimu ścieżki aplikacji procesu (jeśli używana jest ścieżka inna niż **Main**),
- łańcucha „Tip”.

Najnowsze wersje bibliotek narzędziowych są również stosowane podczas testowania iteracyjnego w produkcie Process Designer. Nie są one wdrażane na serwerze produkcyjnym.

W przypadku wdrażania najnowszych wersji biblioteki narzędziowej kontekst wersji jest kombinacją następujących elementów:

- akronimu nazwy biblioteki narzędziowej,
- akronimu ścieżki biblioteki narzędziowej (jeśli używana jest ścieżka inna niż **Main**),
- łańcucha „Tip”.

Przykłady

Zasoby powinny być jednoznacznie nazwane i zewnętrznie identyfikowane przy użyciu kontekstu wersji.

- Poniższa tabela przedstawia przykłady nazw, które są jednoznacznie identyfikowane. W tym przykładzie najnowsza wersja aplikacji procesu używa domyślnej nazwy ścieżki (**Main**):

Tabela 37. Najnowsza wersja aplikacji procesu z domyślną nazwą ścieżki

Typ nazwy	Przykład
Nazwa aplikacji procesu	Aplikacja procesu 1
Akronim nazwy aplikacji procesu	Ap1
Ścieżka aplikacji procesu	Main
Akronim ścieżki aplikacji procesu	"" (jeśli ścieżka to Main)
Obraz stanu aplikacji procesu	
Akronim obrazu stanu aplikacji procesu	Tip

Wszystkie moduły SCA powiązane z tą najnowszą wersją aplikacji procesu zawierają kontekst wersji. Zostało to przedstawione w poniższej tabeli:

Tabela 38. Moduły SCA i pliki EAR z uwzględnieniem wersji

Nazwa modułu SCA	Nazwa z uwzględnieniem wersji	Nazwa pliku EAR / aplikacji z uwzględnieniem wersji
M1	Ap1-Tip-M1	Ap1-Tip-M1.ear
M2	Ap1-Tip-M2	Ap1-Tip-M2.ear

- Poniższa tabela przedstawia przykład najnowszej wersji aplikacji procesu używającej nazwy ścieżki innej niż domyślna:

Tabela 39. Najnowsza wersja aplikacji procesu z nazwą ścieżki inną niż domyślna

Typ nazwy	Przykład
Nazwa aplikacji procesu	Aplikacja procesu 1
Akronim nazwy aplikacji procesu	Ap1
Ścieżka aplikacji procesu	Sciezka1
Akronim ścieżki aplikacji procesu	S1
Obraz stanu aplikacji procesu	
Akronim obrazu stanu aplikacji procesu	Tip

Wszystkie moduły SCA powiązane z tą najnowszą wersją aplikacji procesu zawierają kontekst wersji. Zostało to przedstawione w poniższej tabeli:

Tabela 40. Moduły SCA i pliki EAR z uwzględnieniem wersji

Nazwa modułu SCA	Nazwa z uwzględnieniem wersji	Nazwa pliku EAR / aplikacji z uwzględnieniem wersji
M1	Ap1-S1-Tip-M1	Ap1-S1-Tip-M1.ear
M2	Ap1-S1-Tip-M2	Ap1-S1-Tip-M2.ear

Podobne konwencje nazewnictwa mają zastosowanie w przypadku wdrożeń zaawansowanych bibliotek narzędziowych i obrazów stanu. Są również stosowane w zaawansowanych obrazach stanu instalowanych na serwerze Process Server.

- Poniższa tabela przedstawia przykłady nazw, które są jednoznacznie identyfikowane. W tym przykładzie obraz stanu aplikacji procesu używa domyślnej nazwy ścieżki (**Main**):

Tabela 41. Obraz stanu aplikacji procesu z domyślną nazwą ścieżki

Typ nazwy	Przykład
Nazwa aplikacji procesu	Aplikacja procesu 1
Akronim nazwy aplikacji procesu	Ap1
Ścieżka aplikacji procesu	Main
Akronim ścieżki aplikacji procesu	"" (jeśli ścieżka to Main)
Obraz stanu aplikacji procesu	Process Shapshot V1
Akronim obrazu stanu aplikacji procesu	PSV1

Wszystkie moduły SCA powiązane z tym obrazem stanu aplikacji procesu zawierają kontekst wersji. Zostało to przedstawione w poniższej tabeli:

Tabela 42. Moduły SCA i pliki EAR z uwzględnieniem wersji

Nazwa modułu SCA	Nazwa z uwzględnieniem wersji	Nazwa pliku EAR / aplikacji z uwzględnieniem wersji
M1	PA1-PSV1-M1	PA1-PSV1-M1.ear
M2	PA1-PSV1-M2	PA1-PSV1-M2.ear

- Poniższa tabela przedstawia przykład obrazu stanu aplikacji procesu używającego nazwy ścieżki innej niż domyślna:

Tabela 43. Obraz stanu aplikacji procesu z nazwą ścieżki inną niż domyślna

Typ nazwy	Przykład
Nazwa aplikacji procesu	Aplikacja procesu 1
Akronim nazwy aplikacji procesu	Ap1
Ścieżka aplikacji procesu	Sciezka1

Tabela 43. Obraz stanu aplikacji procesu z nazwą ścieżki inną niż domyślna (kontynuacja)

Typ nazwy	Przykład
Akronim ścieżki aplikacji procesu	S1
Obraz stanu aplikacji procesu	Process Snapshot V1
Akronim obrazu stanu aplikacji procesu	PSV1

Wszystkie moduły SCA powiązane z tym obrazem stanu aplikacji procesu zawierają kontekst wersji. Zostało to przedstawione w poniższej tabeli:

Tabela 44. Moduły SCA i pliki EAR z uwzględnieniem wersji

Nazwa modułu SCA	Nazwa z uwzględnieniem wersji	Nazwa pliku EAR / aplikacji z uwzględnieniem wersji
M1	PA1-T1-PSV1-M1	PA1-T1-PSV1-M1.ear
M2	PA1-T1-PSV1-M2	PA1-T1-PSV1-M2.ear

Konwencje nazewnictwa dla wdrożeń na serwerze Process Server

Istnieje możliwość wdrożenia obrazu stanu aplikacji procesu na serwerze Process Server. Do jednoznacznego identyfikowania wersji używany jest akronim obrazu stanu aplikacji procesu.

W przypadku wdrażania obrazów stanu aplikacji procesu kontekst wersji jest kombinacją następujących elementów:

- akronimu nazwy aplikacji procesu,
- akronimu obrazu stanu aplikacji procesu.

Zasoby powinny być jednoznacznie nazwane i zewnętrznie identyfikowane przy użyciu kontekstu wersji. Poniższa tabela przedstawia przykłady nazw, które są jednoznacznie identyfikowane:

Tabela 45. Przykłady nazw i akronimów

Typ nazwy	Przykład
Nazwa aplikacji procesu	Aplikacja procesu 1
Akronim nazwy aplikacji procesu	Ap1
Obraz stanu aplikacji procesu	1.0.0
Akronim obrazu stanu aplikacji procesu	1.0.0

Elementem identyfikowania zasobu takiego jak moduł lub biblioteka jest kontekst wersji.

Poniższa tabela przedstawia przykład dwóch modułów oraz uwzględniania kontekstu wersji w powiązanych plikach EAR:

Tabela 46. Moduły SCA i pliki EAR z uwzględnieniem wersji

Nazwa modułu SCA	Nazwa z uwzględnieniem wersji	Nazwa pliku EAR / aplikacji z uwzględnieniem wersji
M1	Ap1-1.0.0-M1	Ap1-1.0.0-M1.ear
M2	Ap1-1.0.0-M2	Ap1-1.0.0-M2.ear

Poniższa tabela przedstawia przykład dwóch bibliotek w zasięgu aplikacji procesu oraz uwzględniania kontekstu wersji w powiązanych plikach JAR:

Tabela 47. Biblioteki w zasięgu aplikacji procesu oraz pliki JAR z uwzględnieniem wersji

Nazwa biblioteki SCA w zasięgu aplikacji procesu	Nazwa z uwzględnieniem wersji	Nazwa pliku JAR z uwzględnieniem wersji
Lib1	Ap1-1.0.0-Lib1	Ap1-1.0.0-Lib1.jar
Lib2	Ap1-1.0.0-Lib2	Ap1-1.0.0-Lib2.jar

Powiązania z uwzględnieniem wersji

Aplikacje procesów mogą zawierać moduły SCA zawierające powiązania importu i eksportu. Podczas jednoczesnego wdrażania aplikacji powiązania dla każdej wersji aplikacji muszą być unikalne. Niektóre powiązania są automatycznie aktualizowane podczas wdrażania tak, aby zagwarantować unikalność dla poszczególnych wersji. W innych przypadkach użytkownik musi samodzielnie zaktualizować powiązanie po wdrożeniu, aby zagwarantować jego unikalność.

Powiązanie z *uwzględnieniem wersji* jest w zasięgu konkretnej wersji aplikacji procesu, co gwarantuje jego unikalność dla poszczególnych aplikacji procesów. Poniższe sekcje opisują powiązania, które są automatycznie aktualizowane w celu uwzględnienia wersji, a także wszystkie działania, jakie powinien podjąć użytkownik w czasie wykonywania, jeśli powiązanie nie uwzględnia wersji. Informacje dotyczące kwestii, które powinny być uwzględnione podczas tworzenia modułów, zawiera sekcja “Uwagi na temat używania powiązań”.

SCA

Jeśli powiązania importu i eksportu modułu są zdefiniowane w zasięgu tej samej aplikacji procesu, nazwa elementu docelowego powiązania SCA jest automatycznie zmieniana tak, aby uwzględniała wersję podczas wdrażania.

Jeśli powiązania nie są zdefiniowane w zasięgu tej samej aplikacji procesu, rejestrowany jest komunikat informacyjny. Użytkownik musi zmodyfikować powiązanie importu po wdrożeniu, zmieniając adres docelowy punktu końcowego. Do zmiany adresu docelowego punktu końcowego można użyć Konsoli administracyjnej.

Usługa Web Service (JAX-WS lub JAX-RPC)

Nazwa docelowego adresu punktu końcowego dla powiązania usługi Web Service jest automatycznie zmieniana tak, aby uwzględniała wersje, jeśli spełnione są wszystkie następujące warunki:

- Jeśli użytkownik zastosował domyślną konwencję nazewnictwa dla adresu:
http://ip:port/nazwa_modułuWeb/sca/nazwa_eksportu
- Jeśli adres punktu końcowego jest adresem SOAP/HTTP.
- Jeśli powiązania importu i eksportu dla modułu są zdefiniowane w zasięgu tej samej aplikacji procesu.

Jeśli te warunki nie są spełnione, rejestrowany jest komunikat informacyjny. Działanie, które należy podjąć, zależy od tego w jaki sposób wdrażana jest aplikacja procesu:

- Jeśli aplikacja procesu jest wdrażana jednocześnie, należy ręcznie zmienić nazwę adresu URL punktu końcowego SOAP/HTTP lub nazwę kolejki docelowej SOAP/JMS tak, aby były one unikalne dla poszczególnych wersji aplikacji procesu. Do zmiany adresu docelowego punktu końcowego po wdrożeniu można użyć Konsoli administracyjnej.
- Jeśli wdrażana jest tylko pojedyncza wersja aplikacji procesu, można zignorować ten komunikat.

W przypadku jednoczesnego wdrażania obrazów stanu powiązania usługi Web Service SOAP/JMS działanie, które należy podjąć, zależy od tego w jaki sposób wdrażana jest aplikacja procesu:

- Jeśli import i eksport docelowy znajdują się w tej samej aplikacji procesu, przed opublikowaniem aplikacji procesu w komponencie Process Center i utworzeniem obrazu stanu wykonaj następujące kroki:
 1. Zmień adres URL punktu końcowego dla eksportu. Upewnij się, że miejsce docelowe i fabryka połączeń są unikalne.

2. Zmień adres URL punktu końcowego dla importu tak, aby był on taki sam jak określony dla eksportu w poprzednim kroku.
- Jeśli import i eksport docelowy znajdują się w różnych aplikacjach procesów, wykonaj następujące kroki:
 1. Zmień adres URL punktu końcowego dla eksportu. Upewnij się, że miejsce docelowe i fabryka połączeń są unikalne.
 2. Opublikuj aplikację procesu w komponencie Process Center.
 3. Utwórz obraz stanu.
 4. Wdróż aplikację procesu na serwerze Process Server.
 5. Użyj Konsoli administracyjnej produktu WebSphere w celu zmiany adresu URL punktu końcowego odpowiedniego importu tak, aby był on taki sam jak określony dla eksportu.

HTTP

Nazwa adresu punktu końcowego dla powiązania HTTP jest automatycznie zmieniana tak, aby uwzględniała wersje, jeśli spełnione są wszystkie następujące warunki:

- Jeśli użytkownik zastosował domyślną konwencję nazewnictwa dla adresu:
`http(s)://ip:port/nazwa_modułuWeb/ścieżka_kontekstu_w_eksporcie`
- Jeśli powiązania importu i eksportu dla modułu są zdefiniowane w zasięgu tej samej aplikacji procesu.

Jeśli te warunki nie są spełnione, rejestrowany jest komunikat informacyjny. Działanie, które należy podjąć, zależy od tego w jaki sposób wdrażana jest aplikacja procesu:

- Jeśli aplikacja procesu jest wdrażana jednocześnie, należy ręcznie zmienić nazwę adresu URL punktu końcowego tak, aby była ona unikalna dla poszczególnych wersji aplikacji procesu. Do zmiany adresu docelowego punktu końcowego po wdrożeniu można użyć Konsoli administracyjnej.
- Jeśli wdrażana jest tylko pojedyncza wersja aplikacji procesu, można zignorować ten komunikat.

Powiązania JMS i ogólne powiązania JMS

Powiązania JMS i ogólne powiązania JMS wygenerowane przez system automatycznie uwzględniają wersje.

Uwaga: W przypadku powiązań JMS i ogólnych powiązań JMS zdefiniowanych przez użytkownika, podczas wdrażania nazwy nie są automatycznie zmieniane w celu uwzględniania wersji. Jeśli powiązanie jest zdefiniowane przez użytkownika, należy zmienić nazwy następujących atrybutów tak, aby były unikalne dla poszczególnych wersji aplikacji procesu:

- konfiguracja punktu końcowego,
- kolejka miejsca docelowego odbioru,
- nazwa portu nasłuchiwania (jeśli jest zdefiniowana).

Jeśli punkt końcowy modułu docelowego ulegnie zmianie, należy ustawić zgodne miejsce docelowe Send.

Powiązania MQ/JMS i MQ

Podczas wdrażania powiązań typu MQ/JMS i MQ nazwy nie są automatycznie zmieniane w celu uwzględniania wersji.

Należy zmienić nazwy następujących atrybutów tak, aby były unikalne dla poszczególnych wersji aplikacji procesu:

- konfiguracja punktu końcowego,
- kolejka miejsca docelowego odbioru.

Jeśli punkt końcowy modułu docelowego ulegnie zmianie, należy ustawić zgodne miejsce docelowe Send.

EJB

Podczas wdrażania powiązań typu EJB nazwy nie są automatycznie zmieniane w celu uwzględnienia wersji.

Należy zmienić atrybut nazw JNDI tak, aby był unikalny dla poszczególnych wersji aplikacji procesu.

Należy zauważyć, że aplikacje klienckie również muszą zostać zaktualizowane tak, aby używały nowych nazw JNDI.

EIS

Nazwa adaptera zasobu jest automatycznie zmieniana podczas wdrażania w celu uwzględnienia wersji, pod warunkiem że nie została zmodyfikowana domyślna nazwa zasobu (*nazwa_moduluApp:Adapter Description*).

Jeśli domyślna nazwa zasobu została zmodyfikowana, nazwy adapterów zasobów muszą być unikalne dla poszczególnych wersji aplikacji procesu.

Jeśli nazwy adaptera zasobu nie są unikalne, podczas wdrażania nie jest rejestrowany żaden komunikat informacyjny, który ostrzega użytkownika. Nazwy adapterów zasobów można zmienić ręcznie po wdrożeniu przy użyciu Konsoli administracyjnej.

Wywoływanie dynamiczne z uwzględnieniem wersji

Komponenty przepływu mediacji można skonfigurować tak, aby kierowały komunikaty do punktów końcowych określanych dynamicznie w czasie wykonywania. Podczas tworzenia modułu mediacji należy skonfigurować wyszukiwanie punktów końcowych tak, aby używane było kierowanie z uwzględnieniem wersji.

Jeśli na potrzeby obrazu stanu używany jest styl IBM_VRM (<wersja>.<wydanie>.<modyfikacja>), można wyeksportować plik EAR aplikacji procesu do produktu WebSphere Service Registry and Repository (WSRR). Podczas tworzenia modułu mediacji należy następnie skonfigurować wyszukiwanie punktów końcowych tak, aby używane było kierowanie z uwzględnieniem wersji. Na przykład w polu **Dopasowanie strategii** należy wybrać opcję **Punkt końcowy powrotu odpowiadający ostatniej zgodnej wersji usług SCA opartych na module**, a w polu **Typ powiązania** należy wybrać opcję **SCA**.

Kolejne wersje aplikacji procesu będą wdrażane na serwerze i publikowane w repozytorium WSRR, a wyszukiwanie punktów końcowych modułu mediacji będzie dynamicznie wywoływało ostatnią zgodną wersję punktu końcowego usługi.

Należy zauważyć, że alternatywnym wyjściem jest ustawienie elementu docelowego w polu SMOHeader, a wartość może być przenoszona przez komunikat żądania.

Wdrażanie aplikacji procesów z modułami i projektami Java

Aplikacje procesów mogą zawierać niestandardowe moduły Java EE i projekty Java. Podczas jednoczesnego wdrażania aplikacji niestandardowy moduł Java EE dla każdej wersji aplikacji musi być unikalny.

Należy zauważyć, że niestandardowe moduły Java EE i projekty Java są wdrażane na serwerze, jeśli są wdrażane z modułem SCA, który ma zadeklarowaną zależność od tych modułów lub projektów. Jeśli podczas deklarowania zależności nie zostanie wybrana opcja **Wdróż z modułem** (opcja domyślna), należy ręcznie wdrożyć moduł lub projekt.

Wdrażanie aplikacji procesów z regułami biznesowymi i selektorami

Jeśli użytkownik wdraża wiele wersji aplikacji procesu, która zawiera regułę biznesową lub komponent selektora, należy wiedzieć w jaki sposób poszczególne wersje używają powiązanych metadanych.

Metadane dynamiczne dla reguły biznesowej lub komponentu selektora są definiowane w czasie wykonywania na podstawie nazwy komponentu, docelowej przestrzeni nazw komponentu oraz typu komponentu. Jeśli dwie wersje

aplikacji procesu zawierającej regułę biznesową lub selektor zostaną wdrożone do tego samego środowiska wykonawczego, będą one współużytkowały te same metadane logiki reguły (reguły biznesowej) lub kierowania (selektora).

Aby każda wersja reguły biznesowej lub komponentu selektora w aplikacji procesu mogła używać własnych metadanych dynamicznych (logiki reguły lub kierowania), należy przeprowadzić refaktoryzację docelowej przestrzeni nazw tak, aby była ona unikalna dla każdej wersji aplikacji procesu.

Obiekty konfiguracyjne

Przy użyciu komend AdminConfig narzędzia administracyjnego wiersza komend produktu WebSphere (wsadmin) można uzyskiwać dostęp do bazy danych i właściwości zabezpieczeń produktu IBM Business Process Manager, a także modyfikować je.

Termin *obiekt konfiguracyjny* oznacza obiekt, do którego dostęp jest uzyskiwany za pośrednictwem komend wsadmin AdminConfig. Aby uzyskać więcej informacji, patrz Właściwości konfiguracji zabezpieczeń.

Architektura wdrażania

Architektura wdrażania produktu IBM Business Process Manager składa się z procesów oprogramowania zwanych serwerami, jednostek topologii zwanych węzłami i komórkami oraz repozytorium konfiguracji służącego do przechowywania informacji konfiguracyjnych.

Komórki

W produkcie IBM Business Process Manager *komórki* są grupami logicznymi zawierającymi co najmniej jeden węzeł w sieci rozproszonej.

Komórka jest pojęciem konfiguracyjnym - przeznaczoną dla administratorów metodą tworzenia powiązań logicznych między węzłami. Administratorzy definiują węzły składające się na komórkę zgodnie z konkretnymi kryteriami, które są odpowiednie w danym środowisku organizacyjnym.

Dane konfiguracji administracyjnej są zapisywane w plikach XML. Komórka zachowuje główne pliki konfiguracyjne dla każdego serwera w każdym węźle komórki. Każdy węzeł i każdy serwer mają również własne lokalne pliki konfiguracyjne. Zmiany w węźle lokalnym lub w pliku konfiguracyjnym serwera są tymczasowe, jeśli serwer należy do komórki. Zmiany lokalne nadpisują konfiguracje komórki w czasie, kiedy obowiązują. Zmiany plików konfiguracyjnych głównego serwera i głównego węzła dokonane na poziomie komórki zastępują jakiekolwiek zmiany tymczasowe dokonane na poziomie węzła. Jest to realizowane, kiedy dokumenty konfiguracji komórki są synchronizowane z węzłami. Synchronizacja jest wykonywana w ramach wyznaczonych zdarzeń, na przykład podczas uruchamiania serwera.

Serwery

Serwery udostępniają podstawową funkcjonalność produktu IBM Business Process Manager. Serwery Process Server rozszerzają możliwość obsługi modułów SCA (Service Component Architecture) przez serwer aplikacji. Inne serwery (menedżery wdrażania oraz agenty węzłów) są używane do zarządzania serwerami procesów.

Serwer procesów może być *serwerem autonomicznym* lub *serwerem zarządzanym*. Serwer zarządzany może opcjonalnie być elementem *klastra*. Kolekcja serwerów zarządzanych, klastrów serwerów, a także innego oprogramowania pośredniego jest nazywana *środowiskiem wdrażania*. W środowisku wdrażania każdy zarządzany serwer lub klastr jest skonfigurowany pod kątem realizowania określonej funkcji w środowisku wdrażania (na przykład hosta miejsca docelowego, hosta modułu aplikacji lub serwera infrastruktury Common Event Infrastructure). Serwer autonomiczny jest skonfigurowany na potrzeby udostępniania wszystkich wymaganych funkcji.

Serwery udostępniają środowisko wykonawcze dla modułów SCA, dla zasobów używanych przez te moduły (źródła danych, specyfikacji aktywowania i miejsc docelowych JMS) oraz dla zasobów dostarczanych przez IBM (miejsc docelowych komunikatów, kontenerów produktu Business Process Choreographer i serwerów infrastruktury Common Event Infrastructure).

Agent węzła to agent administracyjny reprezentujący węzeł w systemie użytkownika, który zarządza serwerami w tym węźle. Agenty węzłów monitorują serwery na hoście oraz kierują żądania administracyjne do serwerów. Agent węzła zostaje utworzony po stowarzyszeniu węzła z menedżerem wdrażania.

Menedżer wdrażania to agent administracyjny, który udostępnia centralny widok zarządzania dla wielu serwerów i klastrów.

Serwer autonomiczny jest definiowany przy użyciu profilu autonomicznego. Menedżer wdrażania jest definiowany przy użyciu profilu menedżera wdrażania. Serwery zarządzane są tworzone w *węźle zarządzanym*, który jest definiowany przy użyciu profilu węzła zarządzanego.

Serwery autonomiczne

Serwer autonomiczny udostępnia środowisko do wdrażania modułów SCA w jednym procesie serwera. Ten proces serwera zawiera co najmniej Konsolę administracyjną, miejsce docelowe wdrażania, obsługę przesyłania komunikatów, menedżer reguł procesów biznesowych oraz serwer infrastruktury CEI (Common Event Infrastructure).

Serwer autonomiczny można łatwo skonfigurować. Udostępnia on konsolę Szybki start, za pomocą której można go uruchamiać i zatrzymywać oraz otwierać galerię przykładów i Konsolę administracyjną. Jeśli zostaną zainstalowane przykłady dla produktu IBM Business Process Manager, a następnie zostanie otwarta galeria przykładów, rozwiązanie przykładowe zostanie wdrożone na serwerze autonomicznym. Przy użyciu Konsoli administracyjnej można przeglądać zasoby użyte w tym przykładzie.

Na serwerze autonomicznym można wdrażać własne rozwiązania, ale nie udostępnia on takiej mocy obliczeniowej, skalowalności ani stabilności, które są wymagane w środowisku produkcyjnym. W przypadku środowisk produkcyjnych lepszym rozwiązaniem jest użycie środowiska wdrożenia sieciowego.

Istnieje możliwość rozpoczęcia pracy z serwerem autonomicznym, a następnie dodanie go do środowiska wdrożenia sieciowego przez stowarzyszenie go w komórce menedżera wdrażania, *o ile z tą komórką nie zostały stowarzyszone żadne inne węzły*. W jednej komórce nie można stowarzyszyć wielu serwerów autonomicznych. Aby stowarzyszyć serwer autonomiczny, należy użyć Konsoli administracyjnej menedżera wdrażania lub komendy **addNode**. Podczas stowarzyszenia przy użyciu komendy **addNode** serwer autonomiczny nie może działać.

Serwer autonomiczny jest definiowany przez profil serwera autonomicznego.

Klastry

Klastry to wspólnie zarządzane grupy serwerów (także pod względem podziału obciążenia).

Klaster może zawierać węzły lub pojedyncze serwery aplikacji. Węzeł jest zazwyczaj fizycznym systemem komputerowym z odrębnym adresem IP, na którym działa co najmniej jeden serwer aplikacji. Klastry mogą być grupowane w ramach konfiguracji komórki, co umożliwia wzajemne logiczne powiązanie wielu serwerów i klastrów z różnymi konfiguracjami i aplikacjami, w zależności od wyboru administratora oraz od tego, czy określone rozwiązanie jest odpowiednie w danym środowisku organizacyjnym.

Klastry odpowiadają za równoważenie obciążenia serwerów. Serwery stanowiące część klastra są nazywane elementami klastra. Podczas instalowania aplikacji w klastrze aplikacja ta jest automatycznie instalowana w każdym elemencie klastra.

Ponieważ każdy element klastra zawiera te same aplikacje, istnieje możliwość rozdzielania zadań klienta w zależności od mocy obliczeniowej poszczególnych komputerów. Jest to realizowane poprzez przypisywanie wag do poszczególnych serwerów.

Przypisywanie wag do serwerów w klastrze pozwala zwiększyć wydajność i usprawnić przełączanie awaryjne. Zadania są przypisywane do serwerów, które mają moc obliczeniową wymaganą do wykonania operacji w ramach zadania. Jeśli jeden serwer jest niedostępny i nie może wykonać zadania, zadanie to jest przypisywane do innego elementu klastra. Ta możliwość zmiany przypisania wiąże się z oczywistymi korzyściami w porównaniu z sytuacją, w której działa pojedynczy serwer aplikacji, ponieważ pojedynczy serwer może zostać przeciążony w przypadku zbyt dużej liczby żądań.

Profile

Profil definiuje unikalne środowisko wykonawcze z oddzielnymi plikami komend, plikami konfiguracyjnymi i plikami dzienników. Profile definiują trzy różne typy środowisk w systemach produktu IBM Business Process Manager: serwer autonomiczny, menedżer wdrażania oraz węzeł zarządzany.

Profile pozwalają na uzyskanie w systemie więcej niż jednego środowiska wykonawczego bez konieczności instalowania wielu kopii plików binarnych produktu IBM Business Process Manager.

Należy użyć produktu Program narzędziowy wiersza komend BPMConfig, aby utworzyć profile produktu IBM BPM. Program narzędziowy wiersza komend **manageprofiles** lub jego graficzny interfejs użytkownika, czyli narzędzie Profile Management Tool (PMT), mogą być używane jako alternatywne sposoby tworzenia profili menedżera wdrażania lub profili węzła zarządzanego. Narzędzie PMT nie jest już obsługiwane w przypadku tworzenia profilu autonomicznego.

Uwaga: W przypadku platform rozproszonych każdy profil ma unikalną nazwę. W systemie z/OS wszystkie profile mają nazwę "default". W systemie z/OS nie można zmieniać nazwy, edytować, kopiować ani usuwać profili.

Typy profili

Następujące typy profili produktu IBM BPM są dostępne w produkcie IBM Business Process Manager 8.5:

Profil autonomiczny produktu IBM BPM

Profil autonomiczny definiuje serwery autonomiczne o możliwościach i funkcjach specyficznych dla konfiguracji produktu IBM BPM Express. Możliwe jest utworzenie profilu autonomicznego za pomocą szablonu profilu BPM/BpmServer, który jest obsługiwany tylko w produkcie IBM BPM Express.

Profil menedżera wdrażania produktu IBM BPM

Profil menedżera wdrażania definiuje menedżer wdrażania, który udostępnia pojedynczy interfejs administracyjny logicznej grupie serwerów na jednej lub większej liczbie stacji roboczych. Możliwe jest utworzenie profilu menedżera wdrażania za pomocą szablonu profilu BPM/BpmDmgr, który jest obsługiwany tylko w produktach IBM BPM Standard i IBM BPM Advanced.

Profil węzła zarządzanego produktu IBM BPM

Profil węzła zarządzanego definiuje węzeł zarządzany w przypadku stowarzyszenia węzła z menedżerem wdrażania. Możliwe jest utworzenie profilu węzła zarządzanego za pomocą szablonu profilu BPM/BpmNode, który jest obsługiwany tylko w produktach IBM BPM Standard i IBM BPM Advanced.

Dla każdej konfiguracji produktu IBM BPM dostępne są specyficzne typy profili.

Tabela 48. Dostępne typy profili produktu IBM BPM

Konfiguracja produktu IBM BPM	Typy profili		
	Autonomiczne	Menedżer wdrażania	Węzeł zarządzany
IBM BPMExpress	Tak	Nie	Nie
IBM BPM Standard	Nie	Tak	Tak
IBM BPM Advanced	Nie	Tak	Tak
Środowisko testów jednostkowych dla produktu IBM Integration Designer	Tak	Opcjonalne	Opcjonalne

Katalog profilu

Dla każdego profilu w systemie istnieje osobny katalog zawierający wszystkie pliki tego profilu. Położenie katalogu profilu należy wskazać podczas tworzenia profilu. Domyślnie znajduje się on w podkatalogu `profiles` w katalogu, w którym zainstalowano produkt IBM Business Process Manager. Na przykład: profil `Dmgr` znajduje się w katalogu `C:\Program Files\IBM\WebSphere\ProcServer\profiles\Dmgr`.

Profil domyślny

Pierwszym profilem tworzonym w ramach instalacji produktu IBM Business Process Manager jest *profil domyślny*. Profil domyślny jest domyślnym obiektem docelowym komend wykonywanych z poziomu podkatalogu `bin` w katalogu, w którym zainstalowano produkt IBM Business Process Manager. Jeśli w systemie istnieje tylko jeden profil, wszystkie komendy są wykonywane względem tego profilu. Po utworzeniu dodatkowego profilu można określić, że ma to być profil domyślny.

Uwaga: Profilem domyślnym nie musi być profil o nazwie "default".

Rozszerzanie profili

Jeśli istnieje już profil menedżera wdrażania, profil węzła zarządzanego lub profil serwera autonomicznego utworzony dla produktu WebSphere Application Server Network Deployment, profil można *rozszerzyć* o obsługę produktu IBM Business Process Manager, uzupełniając w ten sposób istniejącą funkcję. Przed rozszerzeniem profilu należy zainstalować produkt IBM Business Process Manager. Następnie za pomocą programu narzędziowego wiersza komend **manageprofiles** należy rozszerzyć profil autonomiczny albo użyć narzędzia Profile Management Tool lub programu narzędziowego wiersza komend **manageprofiles** do rozszerzenia profilu menedżera wdrażania lub profilu węzła zarządzanego.

Ważne: W środowisku wdrożenia sieciowego należy w pierwszej kolejności rozszerzyć profil menedżera wdrażania, a następnie rozszerzyć profile węzła zarządzanego.

Ograniczenie: Nie można rozszerzyć profilu autonomicznego lub profilu menedżera wdrażania, w przypadku którego zmieniono domyślny rejestr użytkowników menedżera VMM produktu WebSphere, na przykład na usługę LDAP.

Menedżery wdrażania

Menedżer wdrażania jest serwerem zarządzającym operacjami dla grupy logicznej lub komórki innych serwerów. Menedżer wdrażania stanowi centrum administrowania serwerami i klastrami.

Profil menedżera wdrażania jest pierwszym profilem, który należy utworzyć podczas tworzenia środowiska wdrażania. Każde utworzone środowisko wdrażania udostępnia konsolę Szybki start, za pomocą której można uruchamiać i zatrzymywać menedżer wdrażania, a także uruchamiać jego Konsolę administracyjną. Konsola administracyjna menedżera wdrażania służy do zarządzania serwerami i klastrami w komórce. Obejmuje to konfigurowanie serwerów i klastrów, dodawanie serwerów do klastrów, uruchamianie i zatrzymywanie serwerów i klastrów oraz wdrażanie modułów SCA.

Choć menedżer wdrażania jest pewnego rodzaju serwerem, nie można instalować modułów bezpośrednio w menedżerze wdrażania.

Węzły

Węzeł jest logiczną grupą serwerów zarządzanych.

Węzeł zazwyczaj odpowiada logicznemu lub fizycznemu systemowi komputerowemu o odrębnym adresie IP hosta. Węzły nie mogą obejmować wielu komputerów. Nazwa węzła jest zwykle taka sama, jak nazwa hosta komputera.

W topologii wdrożenia sieciowego mogą istnieć węzły zarządzane i niezarządzane. W węźle zarządzanym istnieje proces agenta węzła, który zarządza konfiguracją i serwerami węzła. W węzłach niezarządzanych nie ma agenta węzła.

Węzły zarządzane

Węzeł zarządzany jest węzłem, który został stowarzyszony w ramach menedżera wdrażania, i który zawiera agent węzła. Może on również zawierać serwery zarządzane. W węźle zarządzanym można konfigurować i uruchamiać serwery zarządzane.

Serwery skonfigurowane w węźle zarządzanym tworzą zasoby środowiska wdrażania. Te serwery są tworzone, konfigurowane, uruchamiane, zatrzymywane, zarządzane i usuwane za pomocą Konsoli administracyjnej menedżera wdrażania.

Węzeł zarządzany zawiera agent węzła, który zarządza wszystkimi serwerami znajdującymi się w węźle.

Jeśli węzeł jest stowarzyszony, proces agenta węzła jest tworzony automatycznie. Aby można było zarządzać konfiguracją profilu, ten agent węzła musi być uruchomiony. Na przykład podczas wykonywania następujących czynności:

- Uruchamianie i zatrzymywanie procesów serwera.
- Synchronizowanie danych konfiguracji w menedżerze wdrażania z kopią w węźle.

Jednak do działania aplikacji lub konfigurowania zasobów w węźle nie jest konieczne uruchomienie agenta węzła.

Węzeł zarządzany może zawierać jeden lub więcej serwerów, zarządzanych za pomocą menedżera wdrażania. Na serwerach w węźle zarządzanym można wdrażać rozwiązania, ale węzeł zarządzany nie zawiera galerii przykładowych aplikacji. Węzeł zarządzany jest definiowany przez profil węzła zarządzanego i udostępnia konsolę Szybki start.

Węzły niezarządzane

Węzeł niezarządzany nie ma agenta węzła, który zarządza jego serwerami.

Węzły niezarządzane w topologii wdrożenia sieciowego mogą mieć definicje serwerów takich jak serwery WWW, ale nie mogą mieć definicji serwerów aplikacji. Węzły niezarządzane nie mogą być stowarzyszone. Oznacza to, że do węzła niezarządzanego nie można dodać agenta węzła. Innym typem węzła niezarządzanego jest serwer autonomiczny. Menedżer wdrażania nie może zarządzać takim serwerem autonomicznym, ponieważ nie jest on znany w komórce. Serwer autonomiczny może być stowarzyszony. Kiedy jest stowarzyszony, tworzony jest automatycznie agent węzła. Węzeł staje się węzłem zarządzanym w komórce.

Agenty węzłów

Agenty węzłów są agentami administracyjnymi, które kierują żądania administracyjne do serwerów.

Agent węzła jest serwerem, który działa na każdym systemie hosta uczestniczącym w konfiguracji wdrożenia sieciowego. Jest to agent w pełni administracyjny i nie bierze udziału w funkcjach udostępniania aplikacji. Agent węzła udostępnia również inne ważne funkcje administracyjne, takie jak usługi przesyłania plików, synchronizację konfiguracji oraz monitorowanie wydajności.

Uwagi na temat nazewnictwa profili, węzłów, serwerów, hostów i komórek

W niniejszym temacie omówiono zastrzeżone terminy oraz kwestie, które należy uwzględnić podczas nadawania nazwy profilu, węzła, serwera, hosta i komórki (jeśli mają zastosowanie). Ten temat odnosi się do platform rozproszonych.

Uwagi na temat nazewnictwa profilu

Nazwa profilu może być dowolną unikalną nazwą z następującymi ograniczeniami. Podczas tworzenia nazwy profilu nie należy używać żadnych z następujących znaków:

- spacje,
- znaki specjalne, które nie są dozwolone w nazwach katalogów w systemie operacyjnym użytkownika, takie jak *, & lub ?,

- ukośniki (/) lub ukośniki odwrotne (\).

Znaki dwubajtowe są dozwolone.

Windows **Uwagi na temat ścieżki do katalogu:** Ścieżka do katalogu musi być mniejsza lub równa 60 znaków. Liczba znaków w katalogu `ścieżka_do_katalogu_profili\nazwa_profilu` musi być mniejsza lub równa 80 znaków.

Uwaga: Tworząc profil w środowisku Windows, należy stosować krótkie nazwy ścieżek, aby uniknąć przekroczenia limitu 255 znaków w ścieżce, który obowiązuje w systemie Windows.

Uwagi na temat nazewnictwa węzłów, serwerów, hostów i komórek

Nazwy zastrzeżone: należy unikać używania nazw zastrzeżonych jako wartości pól. Używanie nazw zastrzeżonych może prowadzić do nieprzewidywalnych wyników. Następujące słowa są zastrzeżone:

- komórki,
- węzły,
- serwery,
- klastry,
- aplikacje,
- wdrożenia.

Opisy pól na stronach Nazwy węzłów i hostów oraz Nazwy węzłów, hostów i komórek: podczas tworzenia profili należy stosować się do odpowiednich wytycznych dotyczących nazewnictwa.

- Profile serwera autonomicznego
- Profile menedżera wdrażania
- Profile węzła zarządzanego

Tabela 49. Wytyczne dotyczące nazewnictwa dla profili serwera autonomicznego

Nazwa pola	Wartość domyślna	Ograniczenia	Opis
Nazwa węzła	<p>Linux UNIX</p> <p>Windows <code>nazwaSkróconaHosta</code> <i>Węzeł NumerWęzła</i> gdzie:</p> <ul style="list-style-type: none"> • <i>nazwaSkróconaHosta</i> jest nazwą skróconą hosta. • <i>NumerWęzła</i> jest numerem kolejnym rozpoczynającym się od wartości 01. 	Należy unikać używania nazw zastrzeżonych.	Należy wybrać dowolną nazwę. Jeśli planowane jest utworzenie więcej niż jednego serwera w systemie, pomocne w uporządkowaniu instalacji będzie użycie nazwy unikalnej.
Nazwa serwera	<p>Linux UNIX</p> <p>Windows <code>serwer1</code></p>	Dla serwera należy użyć nazwy unikalnej.	Nazwa logiczna serwera.
Nazwa hosta	<p>Linux UNIX</p> <p>Windows Długa postać nazwy serwera nazw domen (DNS).</p>	Należy użyć pełnej nazwy hosta, która jest dostępna w używanej sieci.	Aby umożliwić komunikowanie się ze stacją roboczą, należy użyć jej bieżącej nazwy DNS lub adresu IP. Informacje dodatkowe o nazwie hosta są zawarte poniżej tabeli.

Tabela 50. Wytyczne dotyczące nazewnictwa dla profili menedżera wdrażania

Nazwa pola	Wartość domyślna	Ograniczenia	Opis
Nazwa węzła	<p>Linux UNIX</p> <p>Windows <i>nazwaSkróconaHosta</i></p> <p>Menedżer komórki <i>NumerWęzła</i> gdzie:</p> <ul style="list-style-type: none"> <i>nazwaSkróconaHosta</i> jest nazwą skróconą hosta. <i>NumerWęzła</i> jest numerem kolejnym rozpoczynającym się od wartości 01. 	Należy użyć unikalnej nazwy menedżera wdrażania. Należy unikać używania nazw zastrzeżonych.	Nazwa używana na potrzeby administrowania w ramach komórki menedżera wdrażania.
Nazwa hosta	<p>Linux UNIX</p> <p>Windows Długa postać nazwy serwera nazw domen (DNS).</p>	Należy użyć pełnej nazwy hosta, która jest dostępna w używanej sieci. Należy unikać używania nazw zastrzeżonych.	Aby umożliwić komunikowanie się ze stacją roboczą, należy użyć jej bieżącej nazwy DNS lub adresu IP. Informacje dodatkowe o nazwie hosta są zawarte poniżej tabeli.
Nazwa komórki	<p>Linux UNIX</p> <p>Windows <i>nazwaSkróconaHosta</i></p> <p>Komórka <i>NumerKomórki</i> gdzie:</p> <ul style="list-style-type: none"> <i>nazwaSkróconaHosta</i> jest nazwą skróconą hosta. <i>NumerKomórki</i> jest numerem kolejnym rozpoczynającym się od wartości 01. 	Należy użyć unikalnej nazwy komórki menedżera wdrażania. Nazwa komórki musi być unikalna w każdym przypadku, w którym produkt działa na tej samej fizycznej stacji roboczej lub w tym samym klastrze składającym się ze stacji, na przykład Sysplex. Ponadto nazwa komórki musi być unikalna w każdej sytuacji, w której wymagane jest połączenie sieciowe między jednostkami (między komórkami lub od klienta, który musi się komunikować ze wszystkimi komórkami). Nazwy komórek również muszą być unikalne, jeśli ich przestrzenie nazw mają zostać stowarzyszone. W przeciwnym razie mogą wystąpić takie objawy, jak wyjątek javax.naming.NameNotFoundException. W tym przypadku należy utworzyć komórki o unikalnych nazwach.	Wszystkie węzły stowarzyszone stają się elementami komórki menedżera wdrażania nazwanej na stronie Nazwy węzłów, hostów i komórek narzędzia Profile Management Tool.

Tabela 51. Wytyczne dotyczące nazewnictwa profili węzła zarządzanego

Nazwa pola	Wartość domyślna	Ograniczenia	Opis
Nazwa węzła	<div style="display: flex; justify-content: space-around;"> Linux UNIX </div> <div style="margin-top: 5px;"> Windows </div> <p><i>nazwaSkróconaHosta</i> Węzeł <i>NumberWęzła</i> gdzie:</p> <ul style="list-style-type: none"> <i>nazwaSkróconaHosta</i> jest nazwą skróconą hosta. <i>NumerWęzła</i> jest numerem kolejnym rozpoczynającym się od wartości 01. 	<p>Należy unikać używania nazw zastrzeżonych.</p> <p>W obrębie komórki menedżera wdrażania należy używać nazwy unikalnej.</p>	<p>Nazwa jest używana do administrowania w komórce menedżera wdrażania, do której dodano profil węzła zarządzanego. W obrębie komórki menedżera wdrażania należy używać nazwy unikalnej.</p>
Nazwa hosta	<div style="display: flex; justify-content: space-around;"> Linux UNIX </div> <div style="margin-top: 5px;"> Windows </div> <p>Długa postać nazwy serwera nazw domen (DNS).</p>	<p>Należy użyć pełnej nazwy hosta, która jest dostępna w używanej sieci.</p>	<p>Aby umożliwić komunikowanie się ze stacją roboczą, należy użyć jej bieżącej nazwy DNS lub adresu IP. Informacje dodatkowe o nazwie hosta są zawarte poniżej tabeli.</p>

Uwagi na temat nazwy hosta:

Nazwa hosta jest nazwą sieciową stacji fizycznej, na której jest zainstalowany węzeł. Nazwa hosta musi być tłumaczona na fizyczny węzeł sieci na serwerze. Jeśli serwer na wiele kart sieciowych, nazwa hosta lub adres IP muszą być tłumaczone na jedną z kart sieciowych. Nazwa hosta jest używana przez węzły zdalne do nawiązywania połączenia i komunikowania się z danym węzłem.

IBM Business Process Manager jest zgodny z protokołem IPv4 (Internet Protocol, wersja 4) i IPv6 (Internet Protocol, wersja 6). Wprowadzanie adresów IP w Konsoli administracyjnej lub w innym miejscu można odbywać się w dowolnym z wymienionych formatów. Należy zauważyć, że jeśli w systemie został zaimplementowany protokół IPv6, adres IP należy wprowadzić w formacie IPv6 i na odwrót, jeśli protokół IPv6 nie jest jeszcze dostępny, należy wprowadzać adresy IP w formacie IPv4. Więcej informacji na temat protokołu IPv6 zawiera następujący opis: IPv6.

Następujące wytyczne mogą pomóc w określeniu odpowiedniej nazwy hosta dla stacji roboczej:

- Wybierz nazwę hosta, która jest dostępna dla innych stacji roboczych znajdujących się w sieci.
- Jako nazwy hosta nie używaj identyfikatora ogólnego (localhost).
- Nie próbuj instalować produktów IBM Business Process Manager na serwerze o nazwie hosta zawierającej znaki z zestawu znaków dwubajtowych (DBCS). Znaki DBCS nie są obsługiwane dla nazw hostów.
- W nazwach serwerów unikaj używania znaków podkreślenia (_). Standardy internetowe wymuszają, aby używane nazwy domenowe były zgodne z wymaganiami obowiązującymi dla nazw hostów. Wymagania te zostały opisane w dokumentach Internet Official Protocol Standards RFC 952 i RFC 1123. Nazwy domenowe mogą zawierać tylko litery (wielkie lub małe) oraz cyfry. Nazwy domenowe mogą także zawierać myślniki (-), jeśli nie znajdują się one na końcu nazwy. Znaki podkreślenia (_) w nazwie hosta nie są obsługiwane. Jeśli produkt IBM Business Process Manager został zainstalowany na serwerze, którego nazwa zawiera znak podkreślenia, do czasu zmiany nazwy dostęp do serwera należy uzyskiwać za pomocą jego adresu IP.

Jeśli na tym samym komputerze definiowane są współistniejące węzły o unikanych adresach IP, każdy adres IP należy zdefiniować w tabeli wyszukiwania serwera nazw domen (DNS). Pliki konfiguracyjne dla serwerów nie udostępniają tłumaczenia nazw domenowych dla wielu adresów IP na stacji roboczej z jednym adresem sieciowym.

Wartość określona dla nazwy hosta jest używana jako wartość właściwości `hostname` w dokumentach związanych z konfiguracją. Wartość nazwy hosta należy określić w jednym z następujących formatów:

- łańcuch pełnej nazwy hosta serwerów nazw domen (DNS), na przykład `xmachine.manhattan.ibm.com`,
- łańcuch domyślnej skróconej nazwy hosta DNS, na przykład `xmachine`,
- liczbowy adres IP, na przykład `127.1.255.3`.

Pełna nazwa hosta DNS ma tę zaletę, że jest całkowicie jednoznaczna i elastyczna. Istnieje możliwość zmiany aktualnego adresu IP systemu hosta bez konieczności zmieniania konfiguracji serwera. Określenie wartości dla nazwy hosta jest szczególnie użyteczne, jeśli planowana jest częsta zmiana adresu IP w sytuacji, gdy do przydzielania adresów IP jest używany protokół DHCP (Dynamic Host Configuration Protocol). Wadą tego formatu jest uzależnienie od systemu DNS. Jeśli system DNS nie jest dostępny, połączenie zostanie zerwane.

Skrócona nazwa hosta jest również tłumaczona dynamicznie. Dla formatu nazwy skróconej istnieje dodatkowa możliwość jego ponownego zdefiniowania w lokalnym pliku hosts, więc system może uruchomić serwer nawet wtedy, gdy jest on odłączony od sieci. Aby można było uruchomić serwer po odłączeniu od sieci, nazwę skróconą należy zdefiniować na wartość 127.0.0.1 (lokalna pętla zwrotna) w pliku hosts. Wadą formatu nazwy skróconej jest uzależnienie od systemu DNS w przypadku dostępu zdalnego. Jeśli system DNS nie jest dostępny, połączenie zostanie zerwane.

Liczbowy adres IP ma tę zaletę, że nie wymaga tłumaczenia nazw przez system DNS. Węzeł zdalny może nawiązać połączenie z węzłem nazwanym liczbowym adresem IP, gdy system DNS nie jest dostępny. Wadą tego formatu jest to, że liczbowy adres IP jest stały. Po każdej zmianie adresu IP stacji roboczej należy zmienić ustawienie właściwości hostName w dokumentach konfiguracyjnych. Dlatego też w przypadku używania DHCP lub częstej zmiany adresów IP nie należy korzystać z liczbowego adresu IP. Inną wadą tego formatu jest brak możliwości użycia węzła, jeśli host jest rozłączony z siecią.

Specyfikacja BPMN 2.0

Definicje procesów biznesowych w produkcie IBM Business Process Manager obsługują podklasę Common Executable klasy zgodności Process Modeling specyfikacji BPMN 2.0, która jest związana z modelami, które można uruchomić.

Specyfikacja BPMN (Business Process Model and Notation) jest fundamentalnym standardem dotyczącym procesów w produktach IBM Process Designer oraz IBM Process Center. Diagramy definicji procesów biznesowych (Business Process Definition - BPD) są oparte na specyfikacji BPMN. W tym temacie opisano niektóre sposoby zastosowania specyfikacji BPMN 2.0 w produkcie IBM Business Process Manager. Więcej informacji na temat specyfikacji BPMN można znaleźć na poświęconej jej stronie <http://www.bpmn.org/>.

IBM Business Process Manager obsługuje następujące typy czynności specyfikacji BPMN 2.0:

- Brak (None - czynność abstrakcyjna w specyfikacji BPMN 2.0)
- Zadanie systemowe (System Task - Service Task w specyfikacji BPMN 2.0)
- Czynność użytkownika (User Task)
- Skrypt (Script)
- Zadanie decyzyjne (Decision Task - Business Rule Task w specyfikacji BPMN 2.0)

Zdarzenia pośrednie komunikatu w produkcie IBM BPM udostępniają funkcje podobne do czynności Send Task i Receive Task w specyfikacji BPMN.

Notacja BPMN 2.0

Począwszy od wersji 7.5.1 ikony czynności na diagramach BPD w produkcie Process Designer zgodne ze specyfikacją BPMN 2.0 zostały zebrane na uproszczonej palecie i są wyświetlane w diagramach procesów. Dzięki tym ikonom można stwierdzić, czy dane działanie jest działaniem systemowym, czynnością użytkownika, zadaniem decyzyjnym, skryptem, czy procesem połączonym. Działania znajdujące się w modelach utworzonych z użyciem wcześniejszych wersji produktu także są reprezentowane za pomocą odpowiednich typów i ikon czynności zgodnych ze specyfikacją BPMN 2.0, gdy są wyświetlane w wersji 7.5.1 lub nowszej.

Działania i czynności

W stosunku do poprzednich wersji produktu Process Designer wprowadzono pewne zmiany terminologiczne. Niektóre z nich dotyczą zmiany nazw typów działań.

- Działania usługi (zautomatyzowane) są obecnie zadaniami systemowymi.
- Działania usługi (zadania) na torze innym niż systemowy są obecnie czynnościami użytkownika.
- Działania usługi (zadania) na torze systemowym są obecnie zadaniami decyzyjnymi, jeśli odwołują się do usługi decyzyjnej.
- Działania usługi (zadania) na torze systemowym są obecnie zadaniami systemowymi, jeśli odwołują się do usługi dowolnego rodzaju innego niż usługa decyzyjna.
- Działania Javascript są obecnie działaniami skryptu.
- Zagnieżdżone działania procesu są obecnie procesami połączonymi.
- Działania zewnętrzne z poprzednich wersji produktu Process Designer są dostępne jako zewnętrzne implementacje czynności użytkownika i zadań systemowych.

Bramki

W stosunku do poprzednich wersji nie nastąpiły żadne zmiany notacji dotyczące bramek. Występują jednak trzy zmiany terminologiczne. Bramka decyzyjna jest obecnie *bramką wykluczającą*, prosta bramka podziału lub łączenia jest obecnie *bramką równoległą*, a warunkowa bramka podziału lub łączenia jest obecnie *bramką obejmującą*.

Pojawił się także nowy typ bramki: *bramka zdarzeń*. Bramka zdarzeń reprezentuje punkt rozgałęzienia w procesie, w którym wychodzące z niego alternatywne ścieżki są wybierane na podstawie występującego zdarzenia, a nie w wyniku wartościowania wyrażen z użyciem danych procesu (jak ma to miejsce w przypadku bramek wykluczających i obejmujących). Wybierana ścieżka jest określana na podstawie konkretnego zdarzenia, zwykle odebrania komunikatu.

Zdarzenia nieprzerywające

W specyfikacji BPMN 2.0 dodano notację na potrzeby zdarzeń nieprzerywających. Domyślnie zdarzenie na granicy przerywa działanie, do którego jest przyłączone. W momencie wyzwolenia zdarzenia działanie zostaje zatrzymane, a znacznik przechodzi dalej w kierunku wychodzącego przepływu sekwencji związanej ze zdarzeniem. Gdy zdarzenie jest ustawione jako nieprzerywające, po wyzwoleniu zdarzenia wykonywanie przyłączonego działania jest kontynuowane równoległe, a nowy znacznik zostaje wygenerowany i przekazany dalej w kierunku wychodzącego przepływu sekwencji związanej ze zdarzeniem. Granicą zdarzenia w przypadku zdarzeń nieprzerywających jest linia kreskowana.

Zdarzenia pośrednie przyłączone do działań są przerywającymi zdarzeniami pośrednimi, jeśli zamykają przyłączone do nich działania, lub nieprzerywającymi zdarzeniami pośrednimi, jeśli nie zamykają przyłączonych do nich działań.

Zdarzenie uruchomienia

Specyfikacja BPMN pozwala na pominięcie w modelach procesów symboli zdarzeń uruchomienia i zakończenia. W produkcie Process Designer wymagana jest obecność zdarzeń uruchomienia i zakończenia w modelach procesów.

W produkcie Process Designer dostępne są różne typy zdarzeń uruchomienia:

procesy

- brak
- komunikat
- doraźne

podprocesy

- brak

podprocesy zdarzenia

- błąd
- komunikat
- licznik czasu

Istnieje możliwość zmiany typu zdarzenia uruchomienia przez edycję właściwości zdarzenia. W procesie może występować wiele zdarzeń uruchomienia przez komunikat, ale można używać tylko jednego ogólnego zdarzenia uruchomienia.

Zdarzenia zakończenia

Dostępne są cztery typy zdarzenia zakończenia: *komunikat*, *przerwanie*, *błąd* i *brak*. Typ zdarzenia zakończenia można zmienić.

Kiedy proces nadrzędny wywołuje proces potomny, a proces potomny uruchamia działanie związane ze zdarzeniem przerwania, proces potomny zatrzymuje się, a proces nadrzędny kontynuuje działanie w kolejnych krokach.

Podprocesy

Specyfikacja BPMN definiuje dwa typy podprocesów: osadzone i wielokrotnego użytku. Oba rodzaje można tworzyć w produkcie Process Designer. Procesy osadzone są nazywane po prostu *podprocesami* w produkcie Process Designer, a w wersji 7.5.1 są nowymi obiektami. Podproces wielokrotnego użytku w specyfikacji BPMN jest nazywany *procesem połączonym* w produkcie Process Designer.

Podproces istnieje w obrębie procesu, który go zawiera, i pozwala na grupowanie kroków procesu w celu zmniejszenia złożoności i uporządkowanie diagramu. Za pomocą podprocesów można w jednym działaniu zwinąć wiele kroków. Podproces jest widoczny jedynie dla procesu, w którym jest zdefiniowany. Podproces istnieje w zasięgu procesu wywołującego i ma dostęp do wszystkich zmiennych w obrębie środowiska. Nie stosuje się mechanizmu przekazywania parametrów do i z procesu osadzonego.

Oprócz podprocesów i procesów połączonych w produkcie Process Designer dostępne są podprocesy zdarzeń. Podproces zdarzenia jest to specjalizowany proces przeznaczony do obsługi zdarzeń. Nie jest on połączony z innymi działaniami za pośrednictwem przepływu sekwencji i występuje tylko, jeśli zostanie wyzwolone jego zdarzenie uruchomienia.

Procesy połączone

Podproces wielokrotnego użytku w specyfikacji BPMN jest nazywany *procesem połączonym* w produkcie Process Designer. Jest to proces utworzony poza bieżącym procesem, który może być wywoływany przez bieżący proces. Jest to proces wielokrotnego użytku, ponieważ może być wywoływany także przez inne definicje procesów. Proces połączony definiuje swoje parametry wejściowe oraz wyjściowe i nie ma dostępu do zasięgu ani środowiska procesu wywołującego. Proces połączony jest podobny do procesu zagnieżdżonego dostępnego we wcześniejszych wersjach. Jego działanie nie uległo zmianie. Wcześniejsze procesy zagnieżdżone zostaną poddane migracji do postaci procesów połączonych. Proces połączony wygląda jak podproces o grubym zarysie, który jest podświetlony w oknie komponentu Inspector.

Pętle

Specyfikacja BPMN udostępnia pojęcie działania, które można powtarzać. Działanie może być niepodzielne, co oznacza, że jest powtarzane, lub może być podprocesem obudowującym ciąg powtarzanych kroków. Po rozwinięciu działania powtarzanego widoczne stają się zawarte w nim działania, których uruchamianie jest powtarzane. Warunek jest zawsze wartościowany na początku każdej iteracji pętli. Nie ma możliwości wartościowania na końcu każdej iteracji pętli.

IBM Business Process Manager udostępnia *pętlę obejmującą wiele instancji* uruchamianą skończoną liczbą razy wraz z zawartymi w niej działaniami, które są uruchamiane sekwencyjnie lub równoległe.

Importowanie procesów innych niż procesy BPMN

Istnieje możliwość zaimportowania modeli, które utworzono w produkcie IBM WebSphere Business Modeler, i używania ich w produkcie Process Designer. Informacje na temat importu BPMN 2.0 zawiera sekcja [Odwzorowywanie](#)

elementów produktu IBM WebSphere Business Modeler w konstrukcje produktu IBM Business Process Manager. Można także importować modele zgodne ze specyfikacją BPMN 2.0 utworzone w produktach IBM WebSphere Business Compass, Rational Software Architect lub w innych środowiskach modelowania.

Definicje procesów biznesowych (BPD)

Aby utworzyć model procesu w produkcie IBM Process Designer, należy utworzyć definicję procesu biznesowego (BPD). Definicja procesu biznesowego może być oparta na zaimportowanym modelu BPMN.

Definicja BPD jest modelem procesu wielokrotnego użytku określającym wspólne elementy wszystkich instancji danego modelu procesu w środowisku wykonawczym. Definicja BPD musi obejmować zdarzenie uruchomienia, zdarzenie zakończenia, przynajmniej jeden tor i jedno lub więcej działań. Szczegółowe informacje na temat ograniczeń dotyczących znaków, które obowiązują w definicjach BPD, zawiera sekcja „Konwencje nazewnictwa stosowane w produkcie IBM Process Designer”.

W definicji procesu biznesowego (Business Process Definition - BPD) potrzebny jest tor dla każdego systemu lub każdej grupy użytkowników, uczestniczących w procesie. Tor może być torem uczestnika lub torem systemowym. Można jednak utworzyć definicję BPD, w której działania grupy i systemu są umieszczone na pojedynczym torze, jeśli takie są preferencje użytkownika. Informacje dotyczące tworzenia definicji BPD można znaleźć w temacie „Tworzenie definicji procesu biznesowego (BPD)” dostępnym w sekcji tematów pokrewnych.

Można wyznaczyć dowolną osobę lub grupę, która będzie odpowiedzialna za działania znajdujące się na torze uczestnika. Każdy tworzony tor jest domyślnie przypisywany do grupy Wszyscy użytkownicy. Z tej domyślnej grupy można korzystać przy uruchamianiu i testowaniu definicji BPD w komponencie Inspector. Grupa Wszyscy użytkownicy obejmuje wszystkich użytkowników należących do grupy zabezpieczeń **tw_allusers** (jest to specjalna grupa zabezpieczeń obejmująca automatycznie wszystkich użytkowników systemu).

Tor systemowy zawiera działania obsługiwane przez określony system komponentu IBM Process Center. Każde działanie wymaga implementacji, która definiuje dane działanie oraz ustawia właściwości czynności bądź zadania. W trakcie implementowania programista tworzy usługę lub pisze kod w języku JavaScript niezbędny do wykonania działań w torze systemowym. Informacje dotyczące usług można znaleźć w temacie „Podstawowe informacje o typach usług” dostępnym w sekcji tematów pokrewnych.

Dla każdej tworzonej definicji BPD należy zadeklarować zmienne służące do przechwytywania danych biznesowych przekazywanych między działaniami. Informacje dotyczące implementowania zmiennych można znaleźć w temacie „Zarządzanie zmiennymi i ich odwzorowywanie” dostępnym w sekcji tematów pokrewnych.

Do definicji BPD można dodawać także zdarzenia. Zdarzenia w produkcie IBM BPM mogą być wyzwalane w wyniku minięcia określonej daty zakończenia, w odpowiedzi na wyjątek lub jako reakcja na nadchodzący komunikat. Żądany wyzwalacz determinuje typ zdarzenia, które należy zaimplementować. Szczegółowe informacje na temat dostępnych typów zdarzeń i ich wyzwalaczy zawiera sekcja „Modelowanie zdarzeń”.

Powiązania

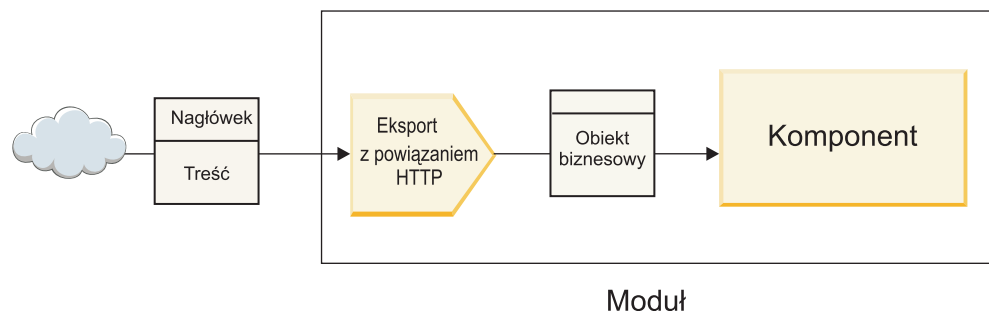
Podstawą architektury zorientowanej na usługi jest pojęcie *usługi* - jednostki funkcjonalności realizowanej poprzez interakcję między urządzeniami komputerowymi. *Eksport* definiuje zewnętrzny interfejs (lub punkt dostępu) modułu, tak aby komponenty architektury SCA (Service Component Architecture) w module mogły udostępniać swoje usługi klientom zewnętrznym. *Import* definiuje interfejs z usługami znajdującymi się poza modulem, aby usługi mogły być wywoływane z wewnątrz modułu. *Powiązania* specyficzne dla protokołu są używane z importami i eksportami w celu określenia sposobu transportowania danych do modułu lub z modułu.

Eksporty

Klienci zewnętrzne mogą wywoływać komponenty SCA w module integracji przy użyciu różnych protokołów (na przykład HTTP, JMS, MQ lub RMI/IIOP) oraz danych w różnych formatach (na przykład XML, CSV, COBOL lub

JavaBeans). Eksporty są komponentami, które odbierają te żądania ze źródeł zewnętrznych, a następnie wywołują komponenty produktu IBM Business Process Manager przy użyciu modelu programistycznego SCA.

Na poniższej ilustracji przedstawiono przykład, w którym eksport odbiera żądanie z aplikacji klienckiej za pośrednictwem protokołu HTTP. Dane są transformowane w obiekt biznesowy - format używany przez komponent SCA. Komponent jest następnie wywoływany przy użyciu tego obiektu danych.

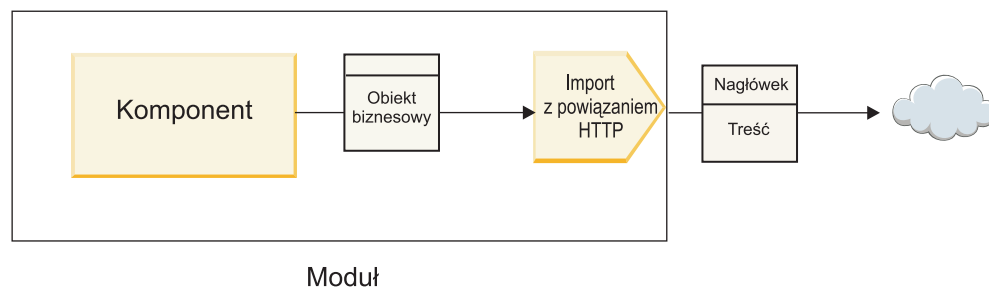


Rysunek 46. Eksport z powiązaniem HTTP

Importy

Użytkownik może wymagać, aby komponent SCA wywoływał usługę zewnętrzną inną niż SCA, która oczekuje danych w innym formacie. Komponent SCA używa importu, aby wywołać usługę zewnętrzną przy użyciu modelu programistycznego SCA. Import następnie wywołuje docelową usługę w sposób oczekiwany przez tę usługę.

Na poniższej ilustracji przedstawiono przykład, w którym żądanie jest wysyłane z komponentu SCA za pośrednictwem importu do usługi zewnętrznej. Obiekt biznesowy (format używany przez komponent SCA) jest transformowany w format oczekiwany przez usługę, a następnie wywoływana jest ta usługa.



Rysunek 47. Import z powiązaniem HTTP

Lista powiązań

Do generowania powiązania dla importu lub eksportu oraz do konfigurowania powiązania używany jest produkt Integration Designer. Poniższa lista opisuje typy dostępnych powiązań.

- SCA
Powiązanie SCA (domyślne) umożliwia usłudze komunikację z usługami w innych modułach SCA. Aby uzyskać dostęp do usługi w innym module SCA, należy użyć importu z powiązaniem SCA. Aby zaoferować usługę innym modułom SCA, należy użyć eksportu z powiązaniem SCA.
- Usługa Web Service
Powiązanie usługi Web Service pozwala uzyskać dostęp do usługi zewnętrznej przy użyciu współdziałających komunikatów SOAP i jakości usług. Powiązań usługi Web Service można także użyć, aby dołączyć załączniki do komunikatu SOAP.

Powiązanie usługi Web Service może używać protokołu transportowego SOAP/HTTP (SOAP przez HTTP) lub SOAP/JMS (SOAP przez JMS). Niezależnie od transportu używanego do przekazania komunikatów SOAP (HTTP lub JMS) powiązania usług Web Services zawsze obsługują interakcję żądanie-odpowiedź w sposób synchroniczny.

- HTTP

Powiązanie HTTP pozwala uzyskać dostęp do usługi zewnętrznej przy użyciu protokołu HTTP w sytuacjach, kiedy używane są komunikaty inne niż SOAP lub kiedy wymagany jest bezpośredni dostęp HTTP. To powiązanie jest używane, jeśli użytkownik pracuje z usługami Web Services opartymi na modelu HTTP (czyli usługami korzystającymi z powszechnie znanych operacji interfejsu HTTP takich jak GET, PUT, DELETE itp.).

- Komponenty EJB (Enterprise JavaBeans)

Powiązania komponentów EJB pozwalają komponentom SCA na interakcję z usługami udostępnianymi przez logikę biznesową Java EE działającą na serwerze Java EE.

- EIS

Powiązanie EIS (Enterprise Information System) używane z adapterem zasobu JCA umożliwia dostęp do usług w systemie EIS lub udostępnienie usług systemowi EIS.

- Powiązania JMS

Powiązania JMS (Java Message Service), ogólne powiązania JMS i powiązania MQ JMS (WebSphere MQ JMS) służą do interakcji z systemami przesyłania komunikatów, w których asynchroniczna komunikacja za pośrednictwem kolejek komunikatów jest niezbędna do zachowania niezawodności.

Eksport z jednym z powiązań JMS obserwuje kolejkę, oczekując nadejścia komunikatu i asynchronicznie wysyła odpowiedź do kolejki odpowiedzi (jeśli odpowiedź jest potrzebna). Import z jednym z powiązań JMS buduje i wysyła komunikat do kolejki JMS i obserwuje kolejkę, oczekując nadejścia odpowiedzi (jeśli odpowiedź jest potrzebna).

- JMS

- Powiązanie JMS umożliwia dostęp do wbudowanego dostawcy JMS produktu WebSphere.

- Ogólne JMS

- Ogólne powiązanie JMS umożliwia dostęp do systemu przesyłania komunikatów od dostawcy innego niż IBM.

- MQ JMS

- Powiązanie MQ JMS umożliwia dostęp do podzbioru JMS systemu przesyłania komunikatów WebSphere MQ. To powiązanie jest używane, jeśli podzbiór funkcji JMS jest wystarczający na potrzeby aplikacji.

- MQ

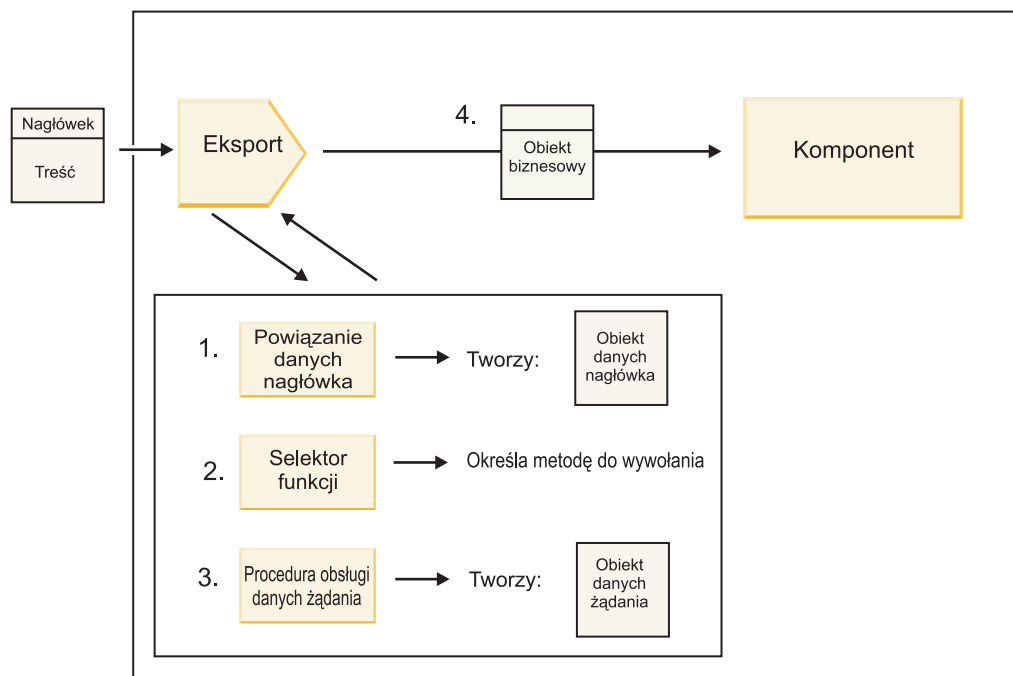
Powiązanie WebSphere MQ umożliwia komunikację z rodzimymi aplikacjami MQ, wprowadzając je do struktury architektury zorientowanej na usługi i zapewniając dostęp do informacji nagłówka specyficznych dla produktu MQ. To powiązanie jest używane, jeśli konieczne jest użycie rodzimych funkcji produktu MQ.

Przegląd powiązań eksportów i importów

Eksport umożliwia udostępnianie usług w module integracji klientom zewnętrznym. Import umożliwia komponentom SCA w module integracji wywoływanie usług wewnętrznych. Powiązanie związane z eksportem lub importem określa relację między komunikatami protokołu i obiektami biznesowymi. Określa również sposób wyboru operacji i błędów.

Przepływ informacji przez eksport

Eksport odbiera żądanie przeznaczone dla komponentu, z którym połączony jest ten eksport za pośrednictwem konkretnego transportu określanego przez związane z eksportem powiązanie (na przykład powiązanie HTTP).



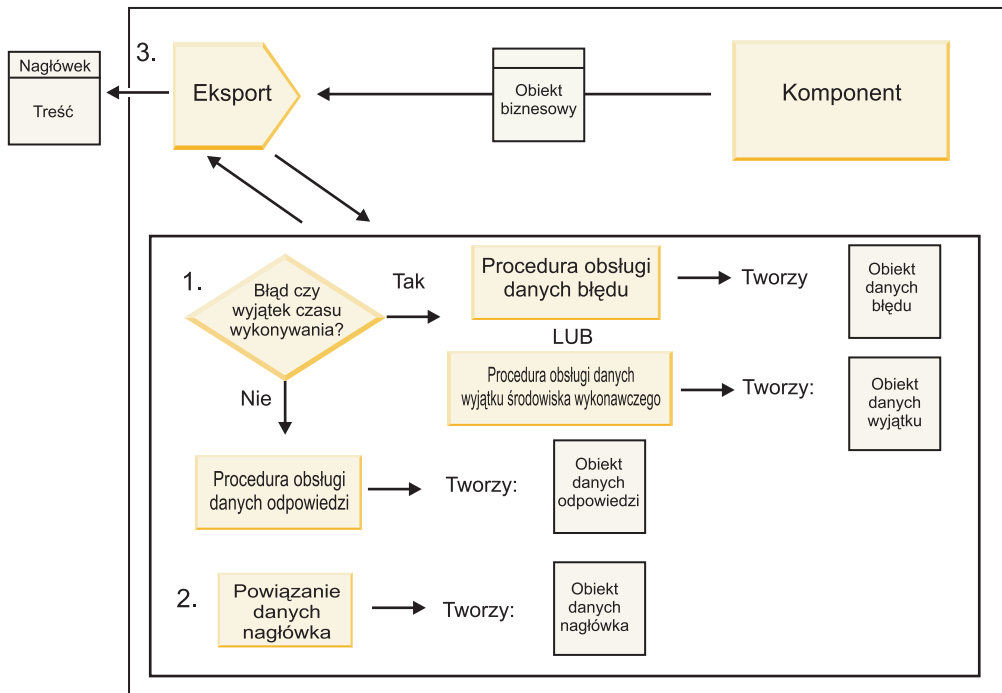
Rysunek 48. Przepływ żądania przez eksport do komponentu

Kiedy eksport odbierze żądanie, ma miejsce następująca sekwencja zdarzeń:

1. Tylko w przypadku powiązań WebSphere MQ: powiązanie danych nagłówka transformuje nagłówek protokołu w obiekt danych nagłówka.
2. Selektor funkcji określa nazwę metody rodzimej na podstawie komunikatu protokołu. Nazwa metody rodzimej jest odwzorowywana przez konfigurację eksportu na nazwę operacji interfejsu eksportu.
3. Procedura obsługi danych żądania lub powiązanie danych w metodzie transformują żądanie w obiekt biznesowy żądania.
4. Eksport wywołuje metodę komponentu z obiektem biznesowym żądania.
 - Powiązanie eksportu HTTP, powiązanie eksportu komponentu EJB i powiązanie eksportu usługi Web Service wywołują komponent SCA synchronicznie.
 - Powiązanie JMS, powiązanie ogólne JMS, powiązanie MQ JMS i powiązanie WebSphere MQ dla eksportu wywołują komponent SCA asynchronicznie.

Należy zauważyć, że eksport może propagować nagłówki i właściwości użytkownika otrzymane za pośrednictwem protokołu, jeśli włączone jest propagowanie kontekstu. Komponenty połączone z eksportem mogą następnie uzyskiwać dostęp do tych nagłówków i właściwości użytkownika. Więcej informacji zawiera temat “Propagacja” w Centrum informacyjnym produktu WebSphere Integration Developer.

Jeśli jest to operacja dwukierunkowa, komponent zwraca odpowiedź.



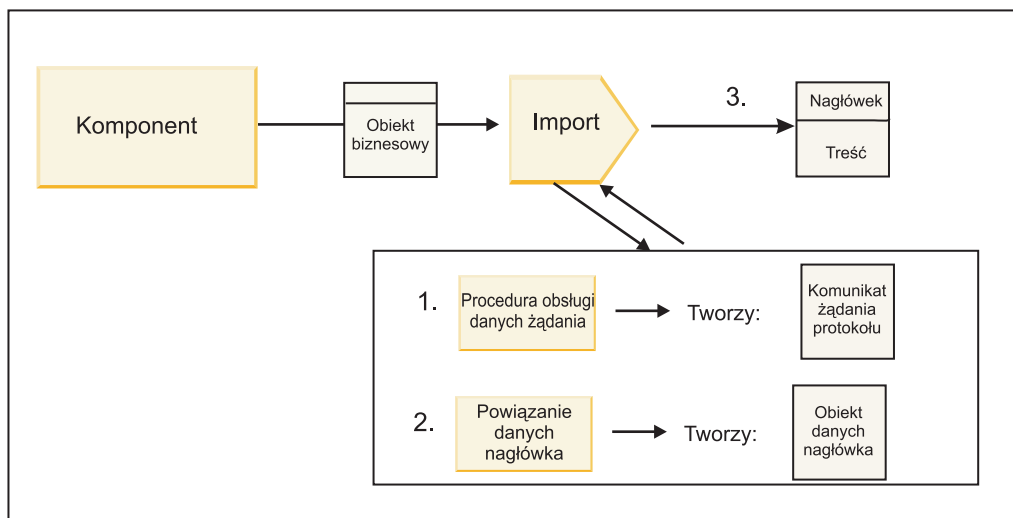
Rysunek 49. Przepływ odpowiedzi z powrotem przez eksport

W tym przypadku ma miejsce następująca sekwencja kroków:

1. Jeśli powiązanie eksportu otrzyma normalny komunikat odpowiedzi, procedura obsługi danych odpowiedzi lub powiązanie danych w metodzie transformują obiekt biznesowy w odpowiedź.
Jeśli odpowiedź jest błędem, procedura obsługi danych błędu lub powiązanie danych w metodzie transformują błąd w odpowiedź błędu.
Tylko w przypadku powiązań eksportu HTTP: jeśli odpowiedź jest wyjątkiem czasu wykonania, wywoływana jest procedura obsługi danych wyjątku czasu wykonania (jeśli jest skonfigurowana).
2. Tylko w przypadku powiązań WebSphere MQ: powiązanie danych nagłówka transformuje obiekty danych nagłówka w nagłówek protokołu.
3. Eksport wysyła odpowiedź usługi przez transport.

Przepływ informacji przez import

Komponenty wysyłają żądania do usług znajdujących się poza modulem za pośrednictwem importu. Żądanie jest wysyłane za pośrednictwem konkretnego transportu, określanego przez związane z importem powiązanie.



Rysunek 50. Przepływ z komponentu, przez import, do usługi

Komponent wywołuje import przy użyciu obiektu biznesowego żądania.

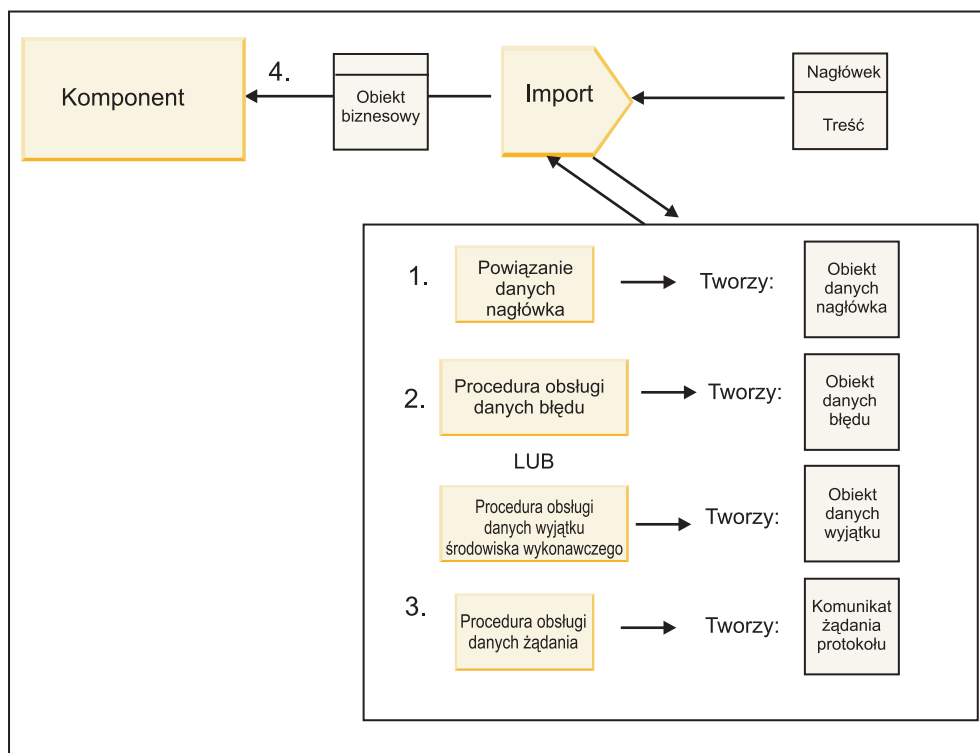
Uwaga:

- Powiązanie importu HTTP, powiązanie importu usługi Web Service i powiązanie importu komponentu EJB powinny być wywoływane synchronicznie przez komponent wywołujący.
- Powiązanie JMS, powiązanie ogólne JMS, powiązanie MQ JMS i powiązanie WebSphere MQ dla importu powinny być wywoływane asynchronicznie.

Po wywołaniu importu przez komponent ma miejsce następująca sekwencja zdarzeń:

1. Procedura obsługi danych żądania lub powiązanie danych w metodzie transformują obiekt biznesowy żądania w komunikat żądania protokołu.
2. Tylko w przypadku powiązań WebSphere MQ: powiązanie danych nagłówka w metodzie ustawia obiekt biznesowy nagłówka w nagłówku protokołu.
3. Import wywołuje usługę przy użyciu żądania usługi za pośrednictwem transportu.

Jeśli jest to operacja dwukierunkowa, usługa zwraca odpowiedź i ma miejsce następująca sekwencja kroków:



Rysunek 51. Przepływ odpowiedzi z powrotem przez import

1. Tylko w przypadku powiązań WebSphere MQ: powiązanie danych nagłówka transformuje nagłówek protokołu w obiekt danych nagłówka.
2. System określa, czy odpowiedź jest błędem.
 - Jeśli odpowiedź jest błędem, selektor błędu bada błąd, aby określić, na jaki błąd WSDL jest on odwzorowywany. Następnie procedura obsługi danych błędu w metodzie transformuje błąd w odpowiedź błędu.
 - Jeśli odpowiedź jest wyjątkiem czasu wykonania, wywoływana jest procedura obsługi danych wyjątku czasu wykonania (jeśli jest skonfigurowana).
3. Procedura obsługi danych odpowiedzi lub powiązanie w metodzie transformują odpowiedź w obiekt biznesowy odpowiedzi.
4. Import zwraca obiekt biznesowy odpowiedzi do komponentu.

Konfiguracja powiązania eksportu i importu

Jednym z najważniejszych aspektów powiązań eksportów i importów jest możliwość transformacji formatu danych, która określa sposób odwzorowania danych (deserializowania) z rodzimego formatu łącznika na obiekt biznesowy lub sposób odwzorowania danych (serializowania) z obiektu biznesowego na rodzimy format łącznika. W przypadku powiązań związanych z eksportami można również określić selektor funkcji, który określa operację, jaka powinna być wykonana na danych. W przypadku powiązań związanych z eksportami lub importami użytkownik może określić sposób obsługi błędów występujących podczas przetwarzania.

Użytkownik określa w powiązaniach również informacje specyficzne dla transportu. Na przykład dla powiązania HTTP określany jest adres URL punktu końcowego. Informacje specyficzne dla transportu, dotyczące powiązania HTTP, są opisane w tematach “Generowanie powiązania importu HTTP” i “Generowanie powiązania eksportu HTTP”. W Centrum informacyjnym można również znaleźć informacje o innych powiązaniach.

Transformacja formatu danych w importach i eksportach

Kiedy w produkcie IBM Integration Designer konfigurowane jest powiązanie eksportu lub importu, jedną z określanych właściwości konfiguracji jest format danych używany przez powiązanie.

- W przypadku powiązań eksportu, w których aplikacja kliencka wysyła żądania do komponentu SCA i odbiera odpowiedzi z tego komponentu, użytkownik wskazuje format danych rodzimych. W zależności od tego formatu system wybiera odpowiednią procedurę obsługi danych lub odpowiednie powiązanie danych, aby transformować dane rodzime na obiekt biznesowy (używany przez komponent SCA), a także aby transformować obiekt biznesowy na dane rodzime (będące odpowiedzią dla aplikacji klienckiej).
- W przypadku powiązań importu, w których komponent SCA wysyła żądania do usługi poza modułem i odbiera odpowiedzi z tej usługi, użytkownik wskazuje format danych dla danych rodzimych. W zależności od tego formatu system wybiera odpowiednią procedurę obsługi danych lub odpowiednie powiązanie danych, aby transformować obiekt biznesowy na dane rodzime i odwrotnie.

IBM Business Process Manager udostępnia zbiór predefiniowanych formatów danych i odpowiadających im procedur obsługi danych lub powiązań danych, które obsługują te formaty. Użytkownik może również utworzyć własne niestandardowe procedury obsługi danych i zarejestrować format danych dla tych procedur obsługi danych. Więcej informacji zawiera temat “Tworzenie procedur obsługi danych” w Centrum informacyjnym produktu IBM Integration Designer.

- *Procedury obsługi danych* są niezależne od protokołu i transformują dane z jednego formatu na inny. W produkcie IBM Business Process Manager procedury obsługi danych zazwyczaj transformują dane rodzime (na przykład XML, CSV lub COBOL) na obiekt biznesowy oraz obiekt biznesowy na dane rodzime. Ponieważ procedury są niezależne od protokołu, można ponownie wykorzystywać tę samą procedurę obsługi danych z różnymi powiązaniem eksportu i importu. Można na przykład używać tej samej procedury obsługi danych XML z powiązaniem eksportu lub importu HTTP albo z powiązaniem eksportu lub importu JMS.
- *Powiązania danych* również transformują dane rodzime na obiekty biznesowe (i odwrotnie), ale są one specyficzne dla protokołu. Na przykład powiązania danych HTTP można użyć tylko z powiązaniem importu lub eksportu HTTP. W przeciwieństwie do procedur obsługi danych, powiązania danych HTTP nie można ponownie wykorzystać z powiązaniem importu lub eksportu MQ.

Uwaga: W produkcie IBM Business Process Manager 7.0 trzy powiązania danych HTTP (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML i HTTPServiceGatewayDataBinding) są nieaktualne. Zamiast nich należy używać procedur obsługi danych (jeśli to możliwe).

Jak wspomniano wcześniej, użytkownik może w razie potrzeby tworzyć niestandardowe procedury obsługi danych. Istnieje również możliwość utworzenia niestandardowych powiązań danych, ale zaleca się tworzenie niestandardowych procedur obsługi danych, ponieważ mogą one być używane z wieloma powiązaniem.

Procedury obsługi danych:

Procedury obsługi danych są konfigurowane w połączeniu z powiązaniem eksportu i importu w celu transformowania danych z jednego formatu na inny w sposób niezależny od protokołu. W ramach produktu udostępnianych jest kilka procedur obsługi danych, ale użytkownik może również w razie potrzeby utworzyć własną procedurę obsługi danych. Procedury obsługi danych można powiązać z powiązaniem eksportu lub powiązaniem importu na jednym z dwóch poziomów. Można je powiązać ze wszystkimi operacjami w interfejsie eksportu lub importu albo z konkretną operacją dla żądania lub odpowiedzi.

Predefiniowane procedury obsługi danych

Do określania procedury obsługi danych, która ma być wykorzystywana, służy produkt IBM Integration Designer.

Predefiniowane procedury obsługi danych są wymienione w poniższej tabeli, która opisuje również sposób transformowania danych przychodzących i wychodzących przez procedurę obsługi danych.

Uwaga: Te procedury obsługi danych mogą być używane z powiązaniem JMS, ogólnymi powiązaniem JMS, powiązaniem MQ JMS, powiązaniem WebSphere MQ i powiązaniem HTTP (z wyjątkiem wskazanych przypadków, w których nie można ich użyć).

Szczegółowe informacje zawiera temat “Procedury obsługi danych” w Centrum informacyjnym produktu Integration Designer.

Tabela 52. Predefiniowane procedury obsługi danych

Procedura obsługi danych	Dane rodzime na obiekt biznesowy	Obiekt biznesowy na dane rodzime
ATOM	Analizuje kanały informacyjne ATOM i umieszcza dane w obiekcie biznesowym kanału informacyjnego ATOM.	Serializuje obiekt biznesowy kanału informacyjnego ATOM do danych w kanale informacyjnym ATOM.
Z ogranicznikami	Analizuje dane z ogranicznikami i umieszcza je w obiekcie biznesowym.	Serializuje obiekt biznesowy do postaci danych z ogranicznikami (między innymi CSV).
O stałej długości	Analizuje dane o stałej długości i umieszcza je w obiekcie biznesowym.	Serializuje obiekt biznesowy do danych o stałej długości.
Obsługiwana przez produkt WTX	Deleguje transformację formatu danych do produktu WebSphere Transformation Extender (WTX). Nazwa odwzorowania WTX jest określana przez procedurę obsługi danych.	Deleguje transformację formatu danych do produktu WebSphere Transformation Extender (WTX). Nazwa odwzorowania WTX jest określana przez procedurę obsługi danych.
Obsługiwana przez użytkownika wywołującego produkt WTX	Deleguje transformację formatu danych do produktu WebSphere Transformation Extender (WTX). Nazwa odwzorowania WTX jest podawana przez użytkownika.	Deleguje transformację formatu danych do produktu WebSphere Transformation Extender (WTX). Nazwa odwzorowania WTX jest podawana przez użytkownika.
JAXB	Serializuje komponenty bean języka Java do postaci obiektu biznesowego przy użyciu reguł odwzorowywania zdefiniowanych przez specyfikację Java Architecture for XML Binding (JAXB).	Deserializuje obiekt biznesowy w komponenty bean języka Java przy użyciu reguł odwzorowywania zdefiniowanych przez specyfikację JAXB.
JAXWS Uwaga: Procedura obsługi danych JAXWS może być używana tylko z powiązaniem EJB.	Procedura używana przez powiązanie EJB do transformowania obiektu Java odpowiedzi lub obiektu wyjątku Java w obiekt biznesowy odpowiedzi przy użyciu reguł odwzorowywania zdefiniowanych przez specyfikację Java API for XML Web Services JAX-WS.	Procedura używana przez powiązanie EJB do transformowania obiektu biznesowego na parametry wychodzącej metody Java przy użyciu reguł odwzorowywania zdefiniowanych przez specyfikację JAX-WS.
JSON	Analizuje dane JSON i umieszcza je w obiekcie biznesowym.	Serializuje obiekt biznesowy do danych JSON.
Rodzima treść	Analizuje rodzime bajty, tekst, odwzorowanie, strumień lub obiekt i umieszcza dane w jednym z pięciu podstawowych obiektów biznesowych (tekstu, bajtów, odwzorowania, strumienia lub obiektu).	Transformuje pięć podstawowych obiektów biznesowych w bajty, tekst, odwzorowanie, strumień lub obiekt.
SOAP	Analizuje komunikat SOAP (oraz nagłówki) i umieszcza dane w obiekcie biznesowym.	Serializuje obiekt biznesowy do komunikatu SOAP.
XML	Analizuje dane XML i umieszcza je w obiekcie biznesowym.	Serializuje obiekt biznesowy do danych XML.
Procedura obsługi danych UTF8 XML	Analizuje dane XML kodowane przy użyciu kodowania UTF-8 i umieszcza je w obiekcie biznesowym.	Serializuje obiekt biznesowy do postaci danych XML kodowanych przy użyciu kodowania UTF-8 podczas wysyłania komunikatu.

Tworzenie procedury obsługi danych

Szczegółowe informacje o tworzeniu procedury obsługi danych zawiera temat “Tworzenie procedur obsługi danych” w Centrum informacyjnym produktu Integration Designer.

Powiązania danych:

Powiązania danych są konfigurowane w połączeniu z powiązaniem eksportu i importu w celu transformowania danych z jednego formatu na inny. Powiązania danych są specyficzne dla protokołu. W ramach produktu udostępnianych jest kilka powiązań danych, ale użytkownik może również w razie potrzeby utworzyć własne powiązanie danych. Powiązanie danych można powiązać z powiązaniem eksportu lub powiązaniem importu na jednym z dwóch poziomów. Powiązać je można ze wszystkimi operacjami interfejsu eksportu lub z konkretną operacją żądania lub odpowiedzi.

Do określenia powiązania danych, które ma być wykorzystywane albo do utworzenia własnego powiązania danych należy użyć produktu IBM Integration Designer. Opis procesu tworzenia powiązań danych zawiera sekcja “Przegląd powiązań JMS, MQ JMS i ogólnych powiązań JMS” w Centrum informacyjnym produktu IBM Integration Designer.

Powiązania JMS

Poniższa tabela przedstawia powiązania danych, których można użyć z następującymi powiązaniem:

- Powiązania JMS
- ogólne powiązania JMS,
- powiązania WebSphere MQ JMS.

Tabela zawiera również opis zadań wykonywanych przez powiązania danych.

Tabela 53. Predefiniowane powiązania danych dla powiązań JMS

Powiązanie danych	Dane rodzime na obiekt biznesowy	Obiekt biznesowy na dane rodzime
Serializowany obiekt Java	Transformuje serializowany obiekt Java w obiekt biznesowy (który jest odwzorowany jako typ wejściowy lub typ wyjściowy w pliku WSDL).	Przekształca obiekt biznesowy do serializowanego obiektu Java w komunikacie obiektu JMS.
Opakowane bajty	Wyodrębnia bajty z przychodzącego komunikatu bajtów JMS i opakowuje je w obiekt biznesowy JMSByteBody.	Wyodrębnia bajty z obiektu biznesowego JMSByteBody i opakowuje je w wychodzący komunikat bajtów JMS.
Opakowana pozycja odwzorowania	Wyodrębnia informacje o nazwie, wartości i typie dla każdej pozycji w przychodzącym komunikacie odwzorowania JMS i tworzy listę obiektów biznesowych MapEntry. Następnie opakowuje listę w obiekt biznesowy JMSTextBody.	Wyodrębnia informacje o nazwie, wartości i typie z listy MapEntry w obiekcie biznesowym JMSTextBody i tworzy odpowiednie pozycje w wychodzącym komunikacie odwzorowania JMS.
Opakowany obiekt	Wyodrębnia obiekt z przychodzącego komunikatu obiektu JMS i opakowuje go w obiekt biznesowy JMSObjectBody.	Wyodrębnia obiekt z obiektu biznesowego JMSObjectBody i opakowuje go w wychodzący komunikat obiektu JMS.
Opakowany tekst	Wyodrębnia tekst z przychodzącego komunikatu tekstu JMS i opakowuje go w obiekt biznesowy JMSTextBody.	Wyodrębnia tekst z obiektu biznesowego JMSTextBody i opakowuje go w wychodzący komunikat tekstu JMS.

Powiązania produktu WebSphere MQ

Poniższa tabela przedstawia powiązania danych, których można użyć z powiązaniem WebSphere MQ oraz opisuje czynności wykonywane przez te powiązania danych.

Tabela 54. Predefiniowane powiązania danych dla powiązań WebSphere MQ

Powiązanie danych	Dane rodzime na obiekt biznesowy	Obiekt biznesowy na dane rodzime
Serializowany obiekt Java	Transformuje serializowany obiekt Java pochodzący z komunikatu przychodzącego na obiekt biznesowy (który jest odwzorowany jako typ wejściowy lub typ wyjściowy w pliku WSDL).	Transformuje obiekt biznesowy w serializowany obiekt Java umieszczony w komunikacie wychodzącym.
Opakowane bajty	Wyodrębnia bajty z nieustrukturyzowanego komunikatu bajtowego MQ i opakowuje je w obiekt biznesowy JMSBytesBody.	Wyodrębnia bajty z obiektu biznesowego JMSBytesBody i opakowuje bajty w wychodzący nieustrukturyzowany komunikat bajtowy MQ.
Opakowany tekst	Wyodrębnia tekst z nieustrukturyzowanego komunikatu tekstu MQ i opakowuje go w obiekt biznesowy JMSTextBody.	Wyodrębnia tekst z obiektu biznesowego JMSTextBody i opakowuje go w nieustrukturyzowany komunikat tekstu MQ.
Opakowana pozycja strumienia	Wyodrębnia informacje o nazwie i typie dla każdej pozycji w przychodzącym komunikacie strumienia JMS i tworzy listę obiektów biznesowych StreamEntry. Następnie opakowuje listę w obiekt biznesowy JMSStreamBody.	Wyodrębnia informacje o nazwie i typie z listy StreamEntry w obiekcie biznesowym JMSStreamBody i tworzy odpowiednie pozycje w wychodzącym komunikacie JMSStreamMessage.

Oprócz powiązań danych wymienionych w tabeli Tabela 25 na stronie 60 produkt WebSphere MQ używa również powiązań danych nagłówka. Szczegółowe informacje na ten temat zawiera Centrum informacyjne produktu IBM Integration Designer.

Powiązania HTTP

Poniższa tabela przedstawia powiązania danych, których można użyć z powiązaniem HTTP, oraz opisuje zadania wykonywane przez powiązania danych.

Tabela 55. Predefiniowane powiązania danych dla powiązań HTTP

Powiązanie danych	Dane rodzime na obiekt biznesowy	Obiekt biznesowy na dane rodzime
Opakowane bajty	Wyodrębnia bajty z treści przychodzącego komunikatu HTTP i opakowuje je w obiekt biznesowy HTTPBytes.	Wyodrębnia bajty z obiektu biznesowego HTTPBytes i dodaje je do treści wychodzącego komunikatu HTTP.
Opakowany tekst	Wyodrębnia tekst z treści przychodzącego komunikatu HTTP i opakowuje go w obiekt biznesowy HTTPText.	Wyodrębnia tekst z obiektu biznesowego HTTPText i dodaje go do treści wychodzącego komunikatu HTTP.

Selektory funkcji w powiązaniach eksportu

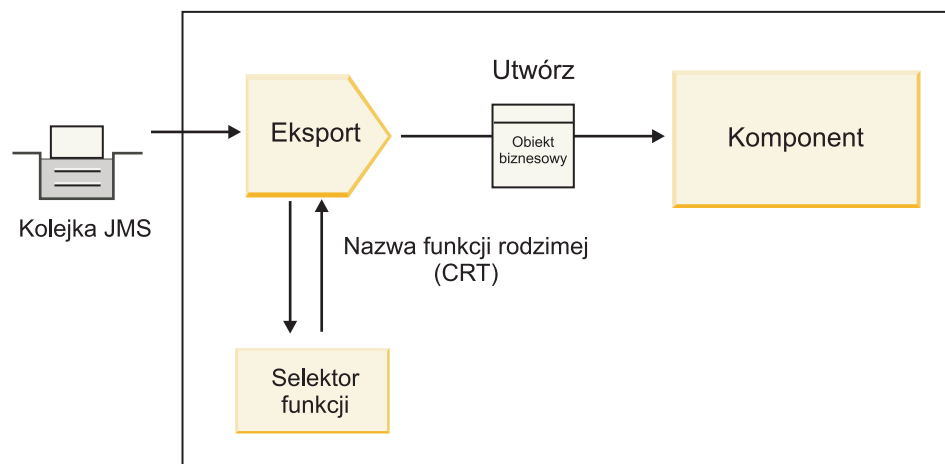
Selektor funkcji służy do wskazania, która operacja powinna zostać wykonana na danych dla określonego komunikatu żądania. Selektory funkcji są konfigurowane w ramach powiązania eksportu.

Rozważmy eksport SCA, który ujawnia interfejs. Interfejs zawiera dwie operacje: tworzenia i aktualizacji. Eksport ma powiązanie JMS, które odczytuje dane z kolejki.

Kiedy komunikat dochodzi do kolejki, do eksportu przekazywane są powiązane dane, ale jak zdecydować, która operacja z interfejsu eksportu powinna być wywołana dla połączonego komponentu? Operacja jest określana przy użyciu selektora funkcji oraz konfiguracji powiązania eksportu.

Selektor funkcji zwraca rodzimą nazwę funkcji (nazwę funkcji w systemie klienta wysyłającym komunikat). Rodzima nazwa funkcji jest następnie odwzorowywana na nazwę operacji lub funkcji w interfejsie powiązonym z eksportem. Na

poniższej przykładowej ilustracji selektor funkcji zwraca rodzimą nazwę funkcji (CRT) z komunikatu przychodzącego, następnie rodzima nazwa funkcji jest odwzorowywana na operację tworzenia, a obiekt biznesowy jest wysyłany do komponentu SCA z operacją tworzenia.



Rysunek 52. Selektor funkcji

Jeśli interfejs ma tylko jedną operację, nie ma potrzeby określania selektora funkcji.

Produkt udostępnia kilka wstępnie spakowanych selektorów funkcji, które są wymienione w kolejnych sekcjach.

Powiązania JMS

Poniższa tabela przedstawia selektory funkcji, których można użyć z następującymi powiązaniem:

- powiązania JMS,
- ogólne powiązania JMS,
- powiązania JMS produktu WebSphere MQ.

Tabela 56. Predefiniowane selektory funkcji dla powiązań JMS

Selektor funkcji	Opis
Selektor funkcji JMS dla prostych powiązań danych JMS	Wybiera nazwę operacji na podstawie właściwości JMSType komunikatu.
Selektor funkcji właściwości nagłówka JMS	Zwraca z nagłówka wartość właściwości łańcucha JMS TargetFunctionName.
Selektor funkcji bramy usług JMS	Określa, czy żądanie jest operacją jednokierunkową czy dwukierunkową, badając właściwość JMSReplyTo ustawioną przez klienta.

Powiązania produktu WebSphere MQ

Poniższa tabela przedstawia selektory funkcji, których można użyć z powiązaniem WebSphere MQ.

Tabela 57. Predefiniowane selektory funkcji dla powiązań WebSphere MQ

Selektor funkcji	Opis
Selektor funkcji handleMessage produktu MQ	Zwraca funkcję handleMessage w postaci wartości, która jest odwzorowywana przy użyciu powiązań metody eksportu na nazwę operacji w interfejsie.

Tabela 57. Predefiniowane selektory funkcji dla powiązań WebSphere MQ (kontynuacja)

Selektor funkcji	Opis
Produkt MQ używa domyślnego selektora funkcji JMS	Odczytuje rodzimą operację z właściwości TargetFunctionName folderu nagłówka MQRFH2.
Produkt MQ używa formatu treści komunikatu jako funkcji rodzimej	Znajduje pole Format ostatniego nagłówka i zwraca to pole w postaci łańcucha.
Selektor funkcji type produktu MQ	Tworzy metodę w powiązaniu eksportu poprzez pobranie adresu URL zawierającego właściwości Msd, Set, Type i Format znajdujące się w nagłówku MQRFH2.
Selektor funkcji bramy usług MQ	Używa właściwości MsgType w nagłówku MQMD do określenia nazwy operacji.

Powiązania HTTP

Poniższa tabela przedstawia selektory funkcji, których można użyć z powiązaniem HTTP.

Tabela 58. Predefiniowane selektory funkcji dla powiązań HTTP

Selektor funkcji	Opis
Selektor funkcji HTTP oparty na nagłówku TargetFunctionName	Używa właściwości nagłówka HTTP TargetFunctionName klienta, aby określić operację, która ma być wywoływana z eksportu w czasie wykonywania.
Selektor funkcji HTTP oparty na metodzie adresu URL i protokołu HTTP	Używa ścieżki względnej z adresu URL, do której dopisana jest metoda HTTP z klienta, aby określić rodzimą operację zdefiniowaną w eksporcie.
Selektor funkcji bramy usług HTTP oparty na metodzie adresu URL z nazwą operacji	Określa metodę wywołania na podstawie adresu URL, jeśli do adresu URL żądania został dopisany łańcuch „operationMode = oneway”.

Uwaga: Istnieje również możliwość utworzenia własnego selektora funkcji przy użyciu produktu IBM Integration Designer. Informacje na temat tworzenia selektora funkcji zawiera Centrum informacyjne produktu IBM Integration Designer. Na przykład sekcja “Przegląd selektorów funkcji MQ” zawiera opis tworzenia selektora funkcji dla powiązań WebSphere MQ.

Obsługa błędów

Użytkownik może skonfigurować powiązania importu i eksportu tak, aby obsługiwać błędy (na przykład wyjątki biznesowe) występujące podczas przetwarzania. Jest to realizowane poprzez określenie procedur obsługi danych błędu. Procedura obsługi danych błędu może być skonfigurowana na trzech poziomach. Procedurę obsługi danych błędu można powiązać z błędem, z operacją lub ze wszystkimi operacjami za pomocą powiązania.

Procedura obsługi danych błędu przetwarza dane błędu i transformuje je na poprawny format w celu wysłania przez powiązanie eksportu lub importu.

- W przypadku powiązania eksportu procedura obsługi danych błędu transformuje obiekt biznesowy wyjątku wysłany z komponentu na komunikat odpowiedzi, który może być używany przez aplikację kliencką.
- W przypadku powiązania importu procedura obsługi danych błędu transformuje dane błędu lub komunikat odpowiedzi wysłany z usługi na obiekt biznesowy wyjątku, który może być użyty przez komponent SCA.

W przypadku powiązań importu powiązanie wywołuje selektor błędu, który określa czy komunikat odpowiedzi jest normalną odpowiedzią, niepowodzeniem biznesowym czy wyjątkiem czasu wykonywania.

Istnieje możliwość określenia procedury obsługi danych błędu dla konkretnego błędu, dla operacji lub dla wszystkich operacji powiązania.

- Jeśli procedura obsługi danych błędu jest skonfigurowana na wszystkich trzech poziomach, wywoływana jest procedura obsługi danych powiązana z określonym błędem.
- Jeśli procedury obsługi danych błędu są skonfigurowane na poziomie operacji i powiązania, wywoływana jest procedura obsługi danych powiązana z operacją.

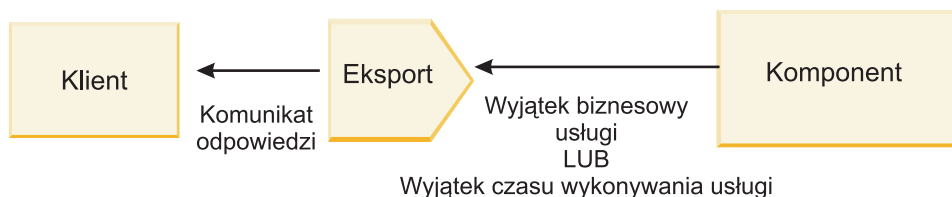
Do określania sposobu obsługi błędów używane są dwa edytory w produkcie IBM Integration Designer. Edytor interfejsów służy do wskazania, czy w operacji będzie występował błąd. Po wygenerowaniu powiązania z tym interfejsem edytor w widoku właściwości umożliwia skonfigurowanie sposobu obsługi błędu. Więcej informacji zawiera temat “Selektory błędów” w Centrum informacyjnym produktu IBM Integration Designer.

Sposób obsługi błędów w powiązaniach eksportu:

Jeśli podczas przetwarzania żądania z aplikacji klienckiej wystąpi błąd, powiązanie eksportu może zwrócić informacje o błędzie klientowi. Użytkownik konfiguruje powiązanie eksportu, określając sposób przetwarzania błędu oraz sposób jego zwracania klientowi.

Powiązanie eksportu jest konfigurowane przy użyciu produktu IBM Integration Designer.

Podczas przetwarzania żądania klient wywołuje eksport przy użyciu żądania, a eksport wywołuje komponent SCA. Podczas przetwarzania żądania komponent SCA może zwrócić odpowiedź biznesową lub zgłosić wyjątek biznesowy usługi albo wyjątek czasu wykonywania usługi. W takiej sytuacji powiązanie eksportu transformuje wyjątek na komunikat błędu i wysyła ten komunikat do klienta, jak przedstawiono na poniższej ilustracji i jak opisano w kolejnych sekcjach.



Rysunek 53. Sposób przesyłania informacji o błędzie z komponentu, przez powiązanie eksportu, do klienta

Użytkownik może utworzyć niestandardową procedurę obsługi danych lub niestandardowe powiązanie danych do obsługi błędów.

Niepowodzenia biznesowe

Niepowodzenia biznesowe są błędami lub wyjątkami biznesowymi, które występują podczas przetwarzania.

Rozważmy następujący interfejs, na którym wykonywana jest operacja createCustomer (tworzenie klienta). Ta operacja ma zdefiniowane dwa niepowodzenia biznesowe: CustomerAlreadyExists (klient już istnieje) i MissingCustomerId (brak identyfikatora klienta).

▼ Operacje

Operacje i ich parametry

	Nazwa	Typ
▼ createCustomer		
▶ Dane wejściowe	Dane wejściowe	Informacje o kliencie
▶ Dane wyjściowe	Dane wyjściowe	Informacje o kliencie
✘ Błąd	Klient już istnieje	Obiekt biznesowy - klient już istnieje
✘ Błąd	Brak identyfikatora klienta	Obiekt biznesowy - brak identyfikatora klienta

Rysunek 54. Interfejs z dwoma błędami

W tym przykładzie, jeśli klient wysłał żądanie utworzenia klienta (do tego komponentu SCA), a ten klient już istnieje, komponent zgłasza błąd `CustomerAlreadyExists` do eksportu. Eksport musi propagować to niepowodzenie biznesowe z powrotem do klienta wywołującego. W tym celu używa procedury obsługi danych błędów, która jest skonfigurowana w powiązaniu eksportu.

Po odebraniu niepowodzenia biznesowego przez powiązanie eksportu, realizowane są następujące działania przetwarzania:

1. Powiązanie określa, która procedura obsługi danych błędów ma być wywołana w celu obsługi błędów. Jeśli wyjątek biznesowy usługi zawiera nazwę błędów, wywoływana jest procedura obsługi danych błędów skonfigurowana dla błędów. Jeśli wyjątek biznesowy usługi nie zawiera nazwy błędów, nazwa błędów jest uzyskiwana poprzez dopasowywanie typów błędów.
2. Powiązanie wywołuje procedurę obsługi danych błędów przy użyciu obiektu danych z wyjątku biznesowego usługi.
3. Procedura obsługi danych błędów transformuje obiekt danych błędów na komunikat odpowiedzi i zwraca ten komunikat do powiązania eksportu.
4. Eksport zwraca komunikat odpowiedzi klientowi.

Jeśli wyjątek biznesowy usługi zawiera nazwę błędów, wywoływana jest procedura obsługi danych błędów skonfigurowana dla błędów. Jeśli wyjątek biznesowy usługi nie zawiera nazwy błędów, nazwa błędów jest uzyskiwana poprzez dopasowywanie typów błędów.

Wyjątki czasu wykonywania

Wyjątek czasu wykonywania jest wyjątkiem, który występuje w aplikacji SCA podczas przetwarzania żądania nieodpowiadającego niepowodzeniu biznesowemu. W przeciwieństwie do niepowodzeń biznesowych, wyjątki czasu wykonywania nie są zdefiniowane w interfejsie.

W niektórych scenariuszach użytkownik może propagować te wyjątki czasu wykonywania do aplikacji klienckiej, aby aplikacja ta mogła podjąć odpowiednie działania.

Jeśli na przykład klient wysłał żądanie (do komponentu SCA), aby utworzyć klienta, a podczas przetwarzania żądania wystąpi błąd autoryzacji, komponent zgłasza wyjątek czasu wykonywania. Ten wyjątek czasu wykonywania musi być propagowany z powrotem do klienta wywołującego, aby klient ten podjął odpowiednie działanie dotyczące autoryzacji. Jest to realizowane przez procedurę obsługi danych wyjątku czasu wykonywania skonfigurowaną w powiązaniu eksportu.

Uwaga: Procedurę obsługi danych wyjątku czasu wykonywania można skonfigurować tylko dla powiązań HTTP.

Przetwarzanie wyjątku czasu wykonywania jest podobne do przetwarzania niepowodzenia biznesowego. Jeśli skonfigurowana została procedura obsługi danych wyjątku czasu wykonywania, realizowane są następujące działania przetwarzania:

1. Powiązanie eksportu wywołuje odpowiednią procedurę obsługi danych przy użyciu wyjątku czasu wykonywania usługi.
2. Procedura obsługi danych transformuje obiekt danych błędu na komunikat odpowiedzi i zwraca ten komunikat do powiązania eksportu.
3. Eksport zwraca komunikat odpowiedzi klientowi.

Obsługa błędów i obsługa wyjątków czasu wykonywania są działaniami opcjonalnymi. Jeśli użytkownik nie chce propagować błędów lub wyjątków czasu wykonywania do klienta wywołującego, nie należy konfigurować procedury obsługi danych błędu lub procedury obsługi danych wyjątku czasu wykonywania.

Sposób obsługi błędów w powiązaniach importu:

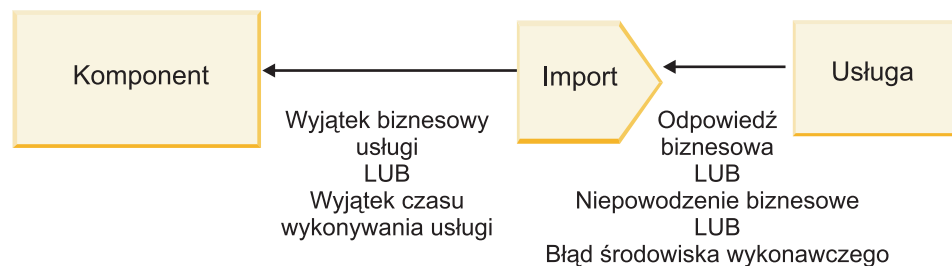
Komponent używa importu, aby wysłać żądanie do usługi znajdującej się poza modulem. Jeśli podczas przetwarzania żądania wystąpi błąd, usługa zwraca błąd do powiązania importu. Użytkownik może skonfigurować powiązanie importu, aby określić sposób przetwarzania błędu oraz sposób jego zwracania komponentowi.

Powiązania importu są konfigurowane przy użyciu produktu IBM Integration Designer. Istnieje możliwość określenia procedury obsługi danych błędu (lub powiązania danych błędu) i selektora błędu.

Procedury obsługi danych błędu

Usługa przetwarzająca żądanie wysyła do powiązania importu informacje o błędzie w postaci wyjątku lub komunikatu odpowiedzi zawierającego dane błędu.

Powiązanie importu transformuje wyjątek usługi lub komunikat odpowiedzi na wyjątek biznesowy usługi lub wyjątek czasu wykonywania usługi, jak przedstawiono na poniższej ilustracji i jak opisano w kolejnych sekcjach.



Rysunek 55. Sposób przesyłania informacji o błędzie z usługi, przez import, do komponentu

Użytkownik może utworzyć niestandardową procedurę obsługi danych lub niestandardowe powiązanie danych do obsługi błędów.

Selektory błędów

Podczas konfigurowania powiązania importu można określić selektor błędu. Selektor błędu określa, czy odpowiedź importu jest rzeczywistą odpowiedzią, wyjątkiem biznesowym czy błędem czasu wykonywania. Określa również, na podstawie treści odpowiedzi lub nagłówka, rodzimą nazwę błędu, która jest odwzorowywana przez konfigurację powiązania na nazwę błędu w powiązany interfejsie.

Z importami JMS, importami MQ JMS, ogólnymi importami JMS, importami WebSphere MQ i importami HTTP mogą być używane dwa typy wstępnie spakowanych selektorów błędów.

Tabela 59. Wstępnie spakowane selektory błędów

Typ selektora błędu	Opis
Oparty na nagłówku	Określa, czy komunikat odpowiedzi jest niepowodzeniem biznesowym, wyjątkiem czasu wykonywania, czy normalnym komunikatem, na podstawie nagłówków w przychodzącym komunikacie odpowiedzi.
SOAP	Określa czy komunikat odpowiedzi SOAP jest normalną odpowiedzią, niepowodzeniem biznesowym, czy wyjątkiem czasu wykonywania.

Poniżej przedstawiono przykłady selektorów błędów opartych na nagłówkach oraz selektora błędu SOAP.

- Selektor błędu oparty na nagłówkach

Jeśli aplikacja ma wskazać, że komunikat przychodzący jest niepowodzeniem biznesowym, w komunikacie przychodzącym muszą występować dwa następujące nagłówki dla niepowodzeń biznesowych:

Header name = FaultType, Header value = Business

Header name = FaultName, Header value = <zdefiniowana przez użytkownika rodzima nazwa błędu>

Jeśli aplikacja ma wskazać, że przychodzący komunikat odpowiedzi jest wyjątkiem czasu wykonywania, w komunikacie przychodzącym musi występować jeden następujący nagłówek:

Header name = FaultType, Header value = Runtime

- Selektor błędów SOAP

Niepowodzenie biznesowe może być wysłane w ramach komunikatu SOAP przy użyciu następującego niestandardowego nagłówka SOAP. W tym przypadku nazwą błędu jest CustomerAlreadyExists.

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

Selektor błędu jest opcjonalny. Jeśli użytkownik nie określi selektora błędu, powiązanie importu nie może określić typu odpowiedzi. W takim przypadku powiązanie traktuje ją jako odpowiedź biznesową i wywołuje procedurę obsługi danych odpowiedzi lub powiązanie danych odpowiedzi.

Istnieje możliwość utworzenia niestandardowego selektora błędu. Kroki tworzenia niestandardowego selektora błędu są przedstawione w temacie “Tworzenie niestandardowego selektora błędu” w Centrum informacyjnym produktu IBM Integration Designer.

Niepowodzenia biznesowe

Niepowodzenie biznesowe może wystąpić w sytuacji, kiedy występuje błąd przetwarzania żądania. Jeśli na przykład użytkownik wysła żądanie utworzenia klienta, a ten klient już istnieje, usługa wysła wyjątek biznesowy do powiązania importu.

Kroki przetwarzania wykonywane po odebraniu wyjątku biznesowego przez powiązanie są zależne od tego, czy dla powiązania został skonfigurowany selektor błędu.

- Jeśli nie został skonfigurowany selektor błędu, powiązanie wywołuje procedurę obsługi danych odpowiedzi lub powiązanie danych odpowiedzi.
- Jeśli został skonfigurowany selektor błędu, realizowane są następujące działania przetwarzania:
 1. Powiązanie importu wywołuje selektor błędu, aby określić, czy odpowiedź jest niepowodzeniem biznesowym, odpowiedzią biznesową czy błędem czasu wykonywania.
 2. Jeśli odpowiedź jest niepowodzeniem biznesowym, powiązanie importu wywołuje selektor błędu, aby udostępnić rodzimą nazwę błędu.
 3. Powiązanie importu określa błąd WSDL odpowiadający rodzimej nazwie błędu zwróconej przez selektor błędu.
 4. Powiązanie importu określa procedurę obsługi danych błędu skonfigurowaną dla tego błędu WSDL.

5. Powiązanie importu wywołuje tę procedurę obsługi danych błędu przy użyciu danych błędu.
6. Procedura obsługi danych błędu transformuje dane błędu na obiekt danych i zwraca ten obiekt do powiązania importu.
7. Powiązanie importu konstruuje obiekt wyjątku biznesowego usługi przy użyciu obiektu danych i nazwy błędu.
8. Import zwraca obiekt wyjątku biznesowego usługi do komponentu.

Wyjątki czasu wykonywania

Wyjątek czasu wykonywania może wystąpić, jeśli istnieje problem dotyczący komunikacji z usługą. Przetwarzanie wyjątku czasu wykonywania jest podobne do przetwarzania wyjątku biznesowego. Jeśli został skonfigurowany selektor błędu, realizowane są następujące działania przetwarzania:

1. Powiązanie importu wywołuje odpowiednią procedurę obsługi danych wyjątku czasu wykonywania przy użyciu danych wyjątku.
2. Procedura obsługi danych wyjątku czasu wykonywania transformuje dane wyjątku na obiekt wyjątku czasu wykonywania usługi i zwraca ten obiekt do powiązania importu.
3. Import zwraca obiekt wyjątku czasu wykonywania usługi do komponentu.

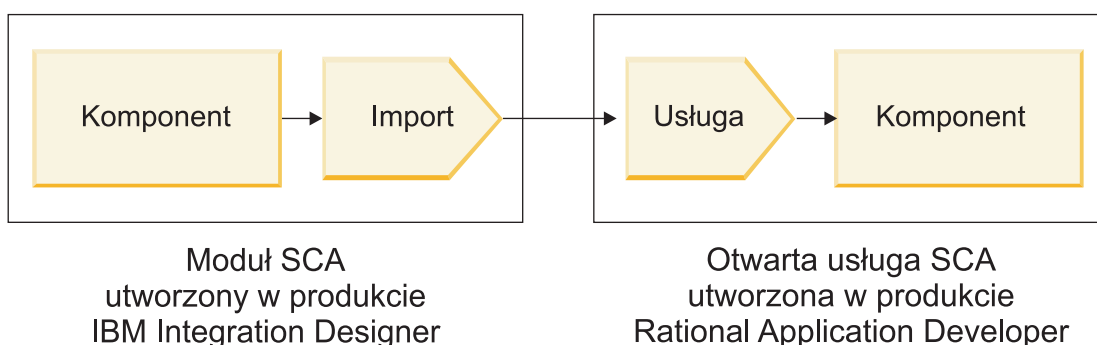
Współdziałanie między modułami SCA i usługami Open SCA

Pakiet składników IBM WebSphere Application Server 7.0 Feature Pack for Service Component Architecture (SCA) udostępnia prosty, ale jednocześnie bogaty w możliwości model programowania do konstruowania aplikacji opartych na specyfikacjach Open SCA. Moduły SCA produktu IBM Business Process Manager używają powiązań importu i eksportu do współdziałania z usługami Open SCA utworzonymi w środowisku Rational Application Developer i udostępnianymi przez pakiet składników WebSphere Application Server Feature Pack for Service Component Architecture.

Aplikacja SCA wywołuje aplikację Open SCA za pośrednictwem powiązania importu. Aplikacja SCA odbiera wywołanie z aplikacji Open SCA za pośrednictwem powiązania eksportu. Listę obsługiwanych powiązań zawiera sekcja “Wywoływanie usług za pośrednictwem powiązań współdziałających” na stronie 69.

Wywoływanie usług Open SCA z modułów SCA

Aplikacje SCA utworzone przy użyciu produktu IBM Integration Designer mogą wywoływać aplikacje Open SCA opracowane w środowisku Rational Application Developer. Niniejsza sekcja zawiera przykład wywoływania usługi Open SCA z poziomu modułu SCA przy użyciu powiązania importu SCA.



Rysunek 56. Komponent w module SCA wywołujący usługę Open SCA

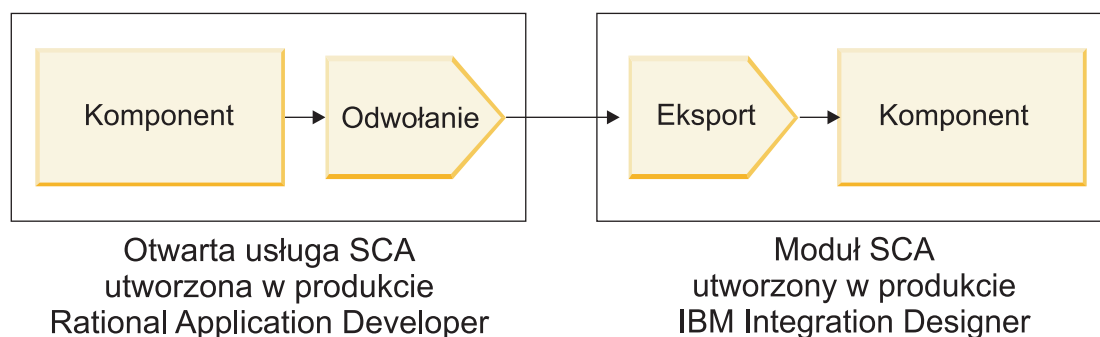
Do wywoływania usługi Open SCA nie jest wymagana żadna specjalna konfiguracja.

Aby połączyć się z usługą Open SCA za pośrednictwem powiązania importu SCA, użytkownik udostępni w powiązaniu importu nazwę komponentu i nazwę usługi Open SCA.

1. Aby uzyskać nazwę komponentu docelowego i usługi z kompozytu Open SCA, wykonaj następujące kroki:
 - a. Upewnij się, czy otwarta jest karta **Właściwości**, wybierając opcję **Okna > Pokaż widok > Właściwości**.
 - b. Otwórz edytor kompozytów, klikając dwukrotnie diagram kompozytu zawierający komponent i usługę. Na przykład: diagramem kompozytu dla komponentu **klient** jest **klient.diagram_kompozytu**.
 - c. Kliknij komponent docelowy.
 - d. Zapisz nazwę komponentu docelowego wyświetlaną w polu **Nazwa** karty **Właściwości**.
 - e. Kliknij ikonę usługi powiązaną z komponentem.
 - f. Zapisz nazwę usługi wyświetlaną w polu **Nazwa** karty **Właściwości**.
2. Aby skonfigurować import produktu IBM Business Process Manager w celu połączenia go z usługą Open SCA, wykonaj następujące kroki:
 - a. W produkcie IBM Integration Designer przejdź do karty **Właściwości** dla importu SCA, który ma zostać połączony z usługą Open SCA.
 - b. W polu **Nazwa modułu** wprowadź nazwę komponentu zapisaną w kroku 1d na stronie 68.
 - c. W polu **Nazwa eksportu** wprowadź nazwę usługi zapisaną w kroku 1f na stronie 68.
 - d. Zapisz pracę, naciskając kombinację klawiszy Ctrl+S.

Wywoływanie modułów SCA z poziomu usług Open SCA

Aplikacje Open SCA utworzone w środowisku Rational Application Developer mogą wywoływać aplikacje SCA utworzone w produkcie IBM Integration Designer. Niniejsza sekcja zawiera przykład wywoływania modułu SCA (za pośrednictwem powiązania eksportu SCA) z poziomu usługi Open SCA.



Rysunek 57. Usługa Open SCA wywołująca komponent w module SCA

Aby nawiązać połączenie z komponentem SCA za pośrednictwem powiązania odwołania Open SCA, należy podać nazwę modułu i nazwę eksportu.

1. Aby uzyskać nazwę modułu docelowego i eksportu, wykonaj następujące kroki:
 - a. Otwórz moduł w edytorze składania produktu IBM Integration Designer, dwukrotnie klikając ten moduł.
 - b. Kliknij eksport.
 - c. Zapisz nazwę eksportu wyświetlaną w polu **Nazwa** karty **Właściwości**.
2. Skonfiguruj odwołanie Open SCA, które ma zostać połączone z modułem i eksportem produktu IBM Business Process Manager:
 - a. Otwórz edytor kompozytów w produkcie Rational Application Developer, dwukrotnie klikając diagram kompozytu zawierający komponent i usługę.
 - b. Kliknij ikonę odwołania dla odwołania komponentu, aby wyświetlić właściwości odwołania na karcie **Właściwości**.
 - c. Kliknij kartę **Powiązanie** umieszczoną po lewej stronie.
 - d. Kliknij opcję **Powiązania**, a następnie opcję **Dodaj**.

- e. Wybierz powiązanie **SCA**.
- f. W polu **Identyfikator URI** wprowadź nazwę modułu produktu IBM Business Process Manager, następnie ukośnik (“/”), a następnie nazwę eksportu (określoną w kroku 1c na stronie 68).
- g. Kliknij przycisk **OK**.
- h. Zapisz pracę, naciskając kombinację klawiszy Ctrl+S.

Wywoływanie usług za pośrednictwem powiązań współdziałających

W celu realizacji współdziałania z usługą Open SCA obsługiwane są następujące powiązania.

- Powiązanie SCA

Jeśli moduł SCA w produkcie IBM Business Process Manager wywołuje usługę Open SCA za pośrednictwem powiązania importu SCA, obsługiwane są następujące style wywoływania:

- asynchroniczny (jednokierunkowy),
- synchroniczny (żądanie-odpowiedź).

Interfejs importu SCA oraz interfejs usługi Open SCA muszą używać interfejsu WSDL zgodnego ze standardem współdziałania usług Web Services (Web Services Interoperability - WS-I).

Uwaga: Powiązanie SCA obsługuje propagację kontekstu transakcji i zabezpieczeń.

- Powiązanie usługi Web Service JAX-WS z protokołem SOAP1.1/HTTP lub SOAP1.2/HTTP

Interfejs importu SCA oraz interfejs usługi Open SCA muszą używać interfejsu WSDL zgodnego ze standardem współdziałania usług Web Services (Web Services Interoperability - WS-I).

Dodatkowo obsługiwane są następujące mechanizmy jakości usług:

- niepodzielne transakcje usług Web Services,
- zabezpieczenia usług Web Services.

- Powiązanie EJB

Do zdefiniowania interakcji między modułem SCA i usługą Open SCA w przypadku używania powiązania EJB stosowany jest interfejs Java.

Uwaga: powiązanie EJB obsługuje propagację kontekstu transakcji i zabezpieczeń.

- Powiązania JMS

Interfejs importu SCA oraz interfejs usługi Open SCA muszą używać interfejsu WSDL zgodnego ze standardem współdziałania usług Web Services (Web Services Interoperability - WS-I).

Obsługiwani są następujący dostawcy JMS:

- przesyłanie komunikatów na platformie WebSphere (powiązanie JMS),
- WebSphere MQ (powiązanie MQ JMS).

Uwaga: Grafy biznesowe nie są obejmowane współdziałaniem w ramach jakichkolwiek powiązań SCA, a więc nie są obsługiwane w interfejsach używanych do współdziałania z pakietem składników WebSphere Application Server Feature Pack for Service Component Architecture.

Typy powiązań

Powiązania specyficzne dla protokołu są używane z importami i eksportami w celu określenia sposobu transportowania danych do modułu lub z modułu.

Wybór odpowiednich powiązań

Tworząc aplikację, należy wiedzieć, w jaki sposób wybrać powiązanie, które najlepiej odpowiada potrzebom danej aplikacji.

Powiązania dostępne w produkcie IBM Integration Designer oferują użytkownikowi kilka opcji do wyboru. Niniejsza lista pomaga określić typ powiązania najlepiej nadający się do realizacji potrzeb danej aplikacji.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Architektura Service Component Architecture (SCA)*:

- Wszystkie usługi są zawarte w modułach (czyli nie ma usług zewnętrznych).
- Użytkownik chce rozdzielić funkcje na różne moduły SCA, które bezpośrednio współdziałają ze sobą.
- Moduły są ściśle sprzężone.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć wykorzystanie powiązania *usługi Web Service*:

- Niezbędny jest dostęp do usługi zewnętrznej za pośrednictwem Internetu lub udostępnienie usługi przez Internet.
- Usługi są luźno sprzężone.
- Preferowana jest komunikacja synchroniczna, czyli żądanie z jednej usługi może oczekiwać na odpowiedź z innej.
- Protokołem usługi zewnętrznej do której uzyskiwany jest dostęp lub usługi, która ma zostać udostępniona, jest protokół SOAP/HTTP lub SOAP/JMS.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć wykorzystanie powiązania *HTTP*:

- Niezbędny jest dostęp do usługi zewnętrznej za pośrednictwem Internetu lub udostępnienie usługi przez Internet, a użytkownik pracuje z innymi usługami Web Services, takimi jak GET, PUT i DELETE.
- Usługi są luźno sprzężone.
- Preferowana jest komunikacja synchroniczna, czyli żądanie z jednej usługi może oczekiwać na odpowiedź z innej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Enterprise JavaBeans (EJB)*:

- Powiązanie jest przeznaczone dla zaimportowanej usługi, która sama jest komponentem EJB lub musi być dostępna za pośrednictwem klientów EJB.
- Zaimportowana usługa jest luźno sprzężona.
- Nie są wymagane stanowe interakcje EJB.
- Preferowana jest komunikacja synchroniczna, czyli żądanie z jednej usługi może oczekiwać na odpowiedź z innej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Enterprise Information Systems (EIS)*:

- Niezbędne jest uzyskanie dostępu do usługi w systemie EIS przy użyciu adaptera zasobu.
- Synchroniczna transmisja danych jest preferowana w odniesieniu do asynchronicznej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Java Message Service (JMS)*:

Ważne: Istnieje kilka typów powiązań JMS. Jeśli użytkownik oczekuje wymiany komunikatów SOAP za pośrednictwem JMS, należy rozważyć użycie powiązania usługi Web Service z protokołem SOAP/JMS. Patrz temat "Powiązania usług Web Services" na stronie 71.

- Niezbędne jest uzyskanie dostępu do systemu przesyłania komunikatów.
- Usługi są luźno sprzężone.
- Asynchroniczna transmisja danych jest preferowana w odniesieniu do synchronicznej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Generic Java Message Service (JMS)*:

- Niezbędne jest uzyskanie dostępu do systemu przesyłania komunikatów od dostawcy innego niż IBM.
- Usługi są luźno sprzężone.
- Niezawodność jest ważniejsza od wydajności, czyli asynchroniczna transmisja danych jest preferowana w odniesieniu do synchronicznej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć użycie powiązania *Message Queue (MQ)*:

- Niezbędny jest dostęp do systemu przesyłania komunikatów WebSphere MQ oraz używania rodzimych funkcji produktu MQ.
- Usługi są luźno sprzężone.

- Niezawodność jest ważniejsza od wydajności, czyli asynchroniczna transmisja danych jest preferowana w odniesieniu do synchronicznej.

Jeśli poniższe czynniki mają zastosowanie, należy rozważyć wykorzystanie powiązania *MQ JMS*:

- Niezbędne jest uzyskanie dostępu do systemu przesyłania komunikatów WebSphere MQ, ale może to być realizowane w kontekście JMS (czyli: podzbiór funkcji JMS jest wystarczający do realizacji potrzeb aplikacji).
- Usługi są luźno sprzężone.
- Niezawodność jest ważniejsza od wydajności, czyli asynchroniczna transmisja danych jest preferowana w odniesieniu do synchronicznej.

Powiązania SCA

Powiązanie SCA (Service Component Architecture) umożliwia usługę komunikację z innymi usługami w innych modułach. Import z powiązaniem SCA pozwala uzyskać dostęp do usługi w innym module SCA. Eksport z powiązaniem SCA pozwala oferować usługę innym modułom.

Do generowania i konfigurowania powiązań SCA w importach i eksportach modułów SCA służy produkt IBM Integration Designer.

Jeśli moduły działają na tym samym serwerze lub są wdrożone w tym samym klastrze, powiązanie SCA jest najłatwiejszym i najszybszym powiązaniem, którego można użyć.

Po wdrożeniu modułu zawierającego powiązanie SCA na serwerze użytkownik może użyć Konsoli administracyjnej, aby wyświetlić informacje o powiązaniu lub, w przypadku powiązania importu, zmienić wybrane właściwości powiązania.

Powiązania usług Web Services

Powiązanie usługi Web Service jest sposobem przesyłania komunikatów z komponentu SCA (Service Component Architecture) do usługi Web Service (i odwrotnie).

Przegląd powiązań usług Web Services: Advanced

Powiązanie importu usługi Web Service umożliwia wywoływanie zewnętrznej usługi Web Service z poziomu komponentów SCA (Service Component Architecture). Powiązanie eksportu usługi Web Service umożliwia ujawnianie klientom komponentów SCA w postaci usług Web Services.

Powiązanie usługi WWW umożliwia uzyskiwanie dostępu do usług zewnętrznych przy użyciu współdziałających komunikatów SOAP i funkcji jakości usług (QoS).

Do generowania i konfigurowania powiązań usług Web Services w importach i eksportach modułów SCA służy produkt Integration Designer. Dostępne są następujące typy powiązań usług Web Service:

- SOAP1.2/HTTP i SOAP1.1/HTTP

Te powiązania są oparte na interfejsie JAX-WS (Java API for XML Web Services) - interfejsie programowania języka Java służącym do tworzenia usług Web Services.

- Jeśli usługa Web Service jest zgodna ze specyfikacją SOAP 1.2, należy użyć powiązania SOAP1.2/HTTP.
- Jeśli usługa Web Service jest zgodna ze specyfikacją SOAP 1.1, należy użyć powiązania SOAP1.1/HTTP.

Ważne: Jeśli wdrażana jest aplikacja z powiązaniem usługi Web Service JAX-WS, docelowy serwer nie może mieć wybranej opcji **Uruchamiaj komponenty, gdy są potrzebne**. Szczegółowe informacje na ten temat zawiera sekcja "Sprawdzanie konfiguracji serwera" na stronie 80.

Po wybraniu jednego z tych powiązań użytkownik może wysłać załączniki z komunikatami SOAP.

Powiązania usług Web Services współpracują ze standardowymi komunikatami SOAP. Korzystając z jednego z powiązań usług Web Services JAX-WS, można jednak dostosować sposób analizowania lub zapisywania komunikatów SOAP. Istnieje na przykład możliwość obsługi niestandardowych elementów w komunikatach SOAP

lub zastosowania dodatkowego przetwarzania komunikatu SOAP. Podczas konfigurowania powiązania użytkownik określa niestandardową procedurę obsługi danych, która realizuje to przetwarzanie komunikatu SOAP.

Istnieje możliwość używania zestawów strategii z powiązaniem usługi Web Service JAX-WS. Zestaw strategii jest kolekcją typów strategii, z których każdy udostępnia jakość usług (QoS). Na przykład: zestaw strategii WSAddressing udostępnia niezależny od transportu sposób jednolitego adresowania usług Web Services i komunikatów. Do wybierania zestawu strategii na potrzeby powiązania służy produkt Integration Designer.

Uwaga: Aby używać zestawu strategii SAML (Security Assertion Markup Language), należy wykonać dodatkowe działania konfiguracyjne opisane w sekcji “Importowanie zestawów strategii SAML” na stronie 77.

- SOAP 1.1/HTTP

Tego powiązania należy użyć, aby utworzyć usługi Web Services używające komunikatów kodowanych SOAP opartych na specyfikacji JAX-RPC (Java API for XML-based RPC).

- SOAP 1.1/JMS

Tego powiązania należy użyć w celu wysyłania lub odbierania komunikatów SOAP przy użyciu miejsca docelowego JMS (Java Message Service).

Niezależnie od transportu używanego do przekazania komunikatów SOAP (HTTP lub JMS) powiązania usług Web Services zawsze obsługują interakcję żądanie-odpowiedź w sposób synchroniczny. Wątek realizujący wywołanie w dostawcy usług jest blokowany do czasu otrzymania odpowiedzi z dostawcy. Więcej informacji na temat tego stylu wywołania zawiera sekcja “Wywołanie synchroniczne”.

Ważne: Dla eksportów w tym samym module nie można używać następujących kombinacji powiązań usług Web Services. Jeśli istnieje potrzeba ujawniania komponentów przy użyciu więcej niż jednego z tych powiązań eksportu, każdy z nich należy umieścić w oddzielnym module, a następnie należy połączyć te moduły z komponentami przy użyciu powiązania SCA:

- SOAP 1.1/JMS i SOAP 1.1/HTTP przy użyciu JAX-RPC,
- SOAP 1.1/HTTP przy użyciu JAX-RPC i SOAP 1.1/HTTP przy użyciu JAX-WS,
- SOAP 1.1/HTTP przy użyciu JAX-RPC i SOAP 1.2/HTTP przy użyciu JAX-WS.

Po wdrożeniu modułu SCA zawierającego powiązanie Web Service na serwerze można użyć Konsoli administracyjnej, aby wyświetlić informacje o powiązaniu lub aby zmienić wybrane właściwości powiązania.

Uwaga: Usługi Web Services umożliwiają współdziałanie aplikacji za pośrednictwem standardowych opisów usług i standardowych formatów wymienianych komunikatów. Powiązania importu i eksportu usług Web Services mogą na przykład współdziałać z usługami zaimplementowanymi przy użyciu standardów WSE (Web Services Enhancements) 3.5 i WCF (Windows Communication Foundation) 3.5 dla platformy Microsoft .NET. W przypadku współdziałania z takimi usługami użytkownik musi zagwarantować, że:

- plik WSDL (Web Services Description Language) używany w celu uzyskania dostępu do eksportu usługi Web Service zawiera niepustą wartość działania SOAP dla każdej operacji w interfejsie,
- klient usługi Web Service ustawia nagłówek SOAPAction lub nagłówek wsa:Action podczas wysyłania komunikatów do eksportu usługi Web Service.

Propagowanie nagłówków SOAP: **Advanced**

Podczas obsługi komunikatów SOAP użytkownik może wymagać dostępu do informacji z niektórych nagłówków SOAP zawartych w odbieranych komunikatach. W takim przypadku upewnij się, że komunikaty z nagłówkami SOAP są wysyłane przy użyciu konkretnych wartości lub opcji przekazywania nagłówków SOAP przez moduł.

Podczas konfigurowania powiązania usługi Web Service w produkcie Integration Designer można wskazać, że nagłówki SOAP mają być propagowane.

- Po odebraniu żądań przez eksport lub odebraniu odpowiedzi przez import istnieje możliwość uzyskania dostępu do informacji w nagłówkach SOAP. Pozwala to oprzeć logikę w module na wartościach nagłówków i umożliwia modyfikowanie tych nagłówków.

- Podczas wysyłania żądań z eksportu lub wysyłania odpowiedzi z importu istnieje możliwość dołączenia nagłówków SOAP do tych komunikatów.

Na formę i obecność propagowanych nagłówków SOAP mogą mieć wpływ zestawy strategii skonfigurowane dla importu lub eksportu. Zostało to wyjaśnione w tabeli Tabela 31 na stronie 74.

Aby skonfigurować propagowanie nagłówków SOAP dla importu lub eksportu, z widoku Właściwości produktu Integration Designer należy wybrać kartę **Propaguj nagłówki protokołu** i zaznaczyć wymagane opcje.

Nagłówek WS-Addressing

Nagłówek WS-Addressing może być propagowany przez powiązanie usługi Web Service JAX-WS.

W przypadku propagowania nagłówka WS-Addressing należy uwzględnić następujące informacje:

- Jeśli zostanie włączone propagowanie nagłówka WS-Addressing, nagłówek ten będzie propagowany do modułu w następujących okolicznościach:
 - podczas odbierania żądań przez eksport,
 - podczas odbierania odpowiedzi przez import.
- Nagłówek WS-Addressing nie jest propagowany do komunikatów wychodzących z produktu IBM Business Process Manager (czyli nagłówek nie jest propagowany podczas wysyłania żądań z importu lub wysyłania odpowiedzi z eksportu).

Nagłówek WS-Security

Nagłówek WS-Security może być propagowany zarówno przez powiązanie usługi Web Service JAX-WS, jak i powiązanie usługi Web Service JAX-RPC.

Specyfikacja WS-Security usług Web Services opisuje udoskonalenia przesyłania komunikatów SOAP, które mają na celu zwiększenie jakości zabezpieczeń za pomocą mechanizmu integralności komunikatów, mechanizmu poufności komunikatów oraz mechanizmu uwierzytelniania pojedynczych komunikatów. Te mechanizmy mogą być używane w połączeniu z wieloma modelami zabezpieczeń i technologiami szyfrowania.

W przypadku propagowania nagłówka WS-Security należy uwzględnić następujące informacje:

- Jeśli zostanie włączone propagowanie nagłówka WS-Security, nagłówek ten będzie propagowany przez moduł w następujących okolicznościach:
 - podczas odbierania żądań przez eksport,
 - podczas wysyłania żądań przez import,
 - podczas odbierania odpowiedzi przez import.
- Nagłówek *nie* będzie domyślnie propagowany podczas wysyłania odpowiedzi przez eksport. Jeśli jednak właściwość maszyny JVM **WSSECURITY.ECHO.ENABLED** zostanie ustawiona na wartość **true**, nagłówek będzie propagowany podczas wysyłania odpowiedzi przez eksport. W tym przypadku, jeśli nagłówek WS-Security w ścieżce żądania nie jest modyfikowany, nagłówki WS-Security mogą być automatycznie kopiowane z żądań do odpowiedzi.
- Dokładna forma komunikatu SOAP wysłanego z importu w przypadku żądania lub z eksportu w przypadku odpowiedzi może nie być w pełni zgodna z pierwotnym, odebrany komunikatem SOAP. Z tego powodu należy założyć, że wszystkie podpisy cyfrowe stają się niepoprawne. Jeśli w wysyłanych komunikatach wymagane jest zastosowanie podpisu cyfrowego, ten podpis należy ustawić przy użyciu odpowiedniego zestawu strategii bezpieczeństwa, a nagłówki WS-Security związane z podpisem cyfrowym i znajdujące się w odebranych komunikatach powinny zostać usunięte wewnątrz modułu.

Aby propagować nagłówek WS-Security, należy dołączyć schemat WS-Security do modułu aplikacji. Procedura dołączania schematu została opisana w sekcji “Uwzględnianie schematu WS-Security w module aplikacji” na stronie 74.

Sposób propagowania nagłówków

Sposób propagowania nagłówków zależy od ustawienia strategii bezpieczeństwa w powiązaniu importu lub eksportu. Zostało to przedstawione w tabeli Tabela 31 na stronie 74:

Tabela 60. Sposób przekazywania nagłówków zabezpieczeń

	Powiązanie eksportu bez strategii bezpieczeństwa	Powiązanie eksportu ze strategią bezpieczeństwa
Powiązanie importu bez strategii bezpieczeństwa	<p>Nagłówki zabezpieczeń są przekazywane wewnątrz modułu bez zmian. Nie są one deszyfrowane.</p> <p>Nagłówki są wysyłane w tej samej formie, w której zostały odebrane.</p> <p>Podpis cyfrowy może stać się niepoprawny.</p>	<p>Nagłówki zabezpieczeń są deszyfrowane i przekazywane przez moduł z uwzględnieniem weryfikacji sygnatury i uwierzytelnienia.</p> <p>Wysyłane są deszyfrowane nagłówki.</p> <p>Podpis cyfrowy może stać się niepoprawny.</p>
Powiązanie importu ze strategią bezpieczeństwa	<p>Nagłówki zabezpieczeń są przekazywane wewnątrz modułu bez zmian. Nie są one deszyfrowane.</p> <p>Nagłówki nie powinny być propagowane do importu. W przeciwnym razie występuje błąd spowodowany duplikowaniem.</p>	<p>Nagłówki zabezpieczeń są deszyfrowane i przekazywane przez moduł z uwzględnieniem weryfikacji sygnatury i uwierzytelnienia.</p> <p>Nagłówki nie powinny być propagowane do importu. W przeciwnym razie występuje błąd spowodowany duplikowaniem.</p>

Należy skonfigurować odpowiednie zestawy strategii dla powiązań eksportu i importu, ponieważ pozwala to odizolować requester usług od zmian w konfiguracji lub wymagań QoS dostawcy usług. Uwidocznienie standardowych nagłówków SOAP w module może następnie posłużyć do wywarcia wpływu na przetwarzanie w module (na przykład rejestrowanie i śledzenie). Propagowanie nagłówków SOAP z odebranego komunikatu przez moduł zmniejsza korzyści płynące z izolowania modułu.

Standardowe nagłówki, takie jak nagłówki WS-Security, nie powinny być propagowane z żądania do importu ani z odpowiedzi do eksportu, jeśli import lub eksport mają powiązany zestaw strategii, który powoduje generowanie tych nagłówków. W przeciwnym razie wystąpi błąd spowodowany duplikowaniem nagłówków. Zamiast tego należy jawnie usunąć nagłówki albo powiązanie importu lub eksportu powinno być skonfigurowane tak, aby uniemożliwić propagowanie nagłówków protokołu.

Uzyskiwanie dostępu do nagłówków SOAP

Jeśli komunikat zawierający nagłówki SOAP zostanie odebrany z importu lub eksportu usługi Web Service, nagłówki są umieszczane w sekcji nagłówków obiektu SMO. Użytkownik może uzyskać dostęp do informacji w nagłówku. Zostało to opisane w sekcji "Uzyskiwanie dostępu do informacji w nagłówku SOAP w obiekcie SMO".

Uwzględnianie schematu WS-Security w module aplikacji

Poniższa procedura opisuje kroki związane z dołączaniem schematu do modułu aplikacji:

- Jeśli komputer, na którym działa produkt Integration Designer, jest podłączony do Internetu, wykonaj następujące kroki:
 1. W perspektywie Integracja biznesowa wybierz opcję **Zależności** dla projektu.
 2. Rozwiń sekcję **Zasoby predefiniowane** i wybierz opcję **Pliki schematu WS-Security 1.0** lub **Pliki schematu WS-Security 1.1**, aby zaimportować schemat do modułu.
 3. Oczyszczyć i odbuduj projekt.

- Jeśli komputer, na którym działa produkt Integration Designer, nie jest podłączony do Internetu, można pobrać schemat na drugi komputer, który jest podłączony do Internetu. Można następnie skopiować te pliki na komputer, na którym działa produkt Integration Designer.

1. Z poziomu komputera podłączonego do Internetu pobierz schemat zdalny:
 - a. Kliknij opcję **Plik > Importuj > Integracja biznesowa > WSDL i XSD**.
 - b. Wybierz opcję **Zdalny plik WSDL** lub **Plik XSD**.
 - c. Zaimportuj następujące schematy:
 - `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. Skopiuj schematy na komputer, który nie jest podłączony do Internetu.
3. Zaimportuj schemat na komputerze, który nie jest podłączony do Internetu:
 - a. Kliknij opcję **Plik > Importuj > Integracja biznesowa > WSDL i XSD**.
 - b. Wybierz opcję **Lokalny plik WSDL** lub **Plik XSD**.
4. Zmień położenia schematów dla pliku `oasis-wss-wssecurity-secext-1.1.xsd`:
 - a. Otwórz schemat znajdujący się w następującym miejscu: *położenie_robocze/nazwa_modułu/StandardImportFilesGen/oasis-wss-wssecurity-secext-1.1.xsd*.
 - b. Zmień element:


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

 na element:


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


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

 na element:


```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
  schemaLocation='../w3/tr/_2002/rec_xmlenc_core_20021210/xenc-schema.xsd' />
```
5. Zmień położenie schematu dla pliku `oasis-200401-wss-wssecurity-secext-1.0.xsd`:
 - a. Otwórz schemat znajdujący się w następującym miejscu: *położenie_robocze/nazwa_modułu/StandardImportFilesGen/oasis-200401-wss-wssecurity-secext-1.0.xsd*.
 - b. Zmień element:


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

 na element:


```
<xsd:import namespace="http://www.w3.org/2000/09/xmlsig#"
  schemaLocation="..w3/tr/_2002/rec_xmlsig_core_20020212/xmlsig-core-schema.xsd"/>
```
6. Oczyszczenie i odbudowa projektu.

Propagowanie nagłówków transportu: Advanced

Podczas obsługi komunikatów SOAP użytkownik może wymagać dostępu do informacji z niektórych nagłówków transportu zawartych w odbieranych komunikatach. W takim przypadku upewnij się, że komunikaty z nagłówkami SOAP są wysyłane przy użyciu konkretnych wartości lub opcji przekazywania nagłówków SOAP przez moduł.

Podczas konfigurowania powiązania usługi Web Service w produkcie Integration Designer można wskazać, że nagłówki transportu mają być propagowane.

- Po odebraniu żądań przez eksport lub odebraniu odpowiedzi przez import istnieje możliwość uzyskania dostępu do informacji w nagłówkach transportu. Pozwala to oprzeć logikę w module na wartościach nagłówków i umożliwia modyfikowanie tych nagłówków.
- Podczas wysyłania odpowiedzi z eksportu lub wysyłania żądań z importu istnieje możliwość dołączenia nagłówków transportu do tych komunikatów.

Określanie propagowania nagłówków

Aby skonfigurować propagowanie nagłówków transportu dla importu lub eksportu, wykonaj następujące kroki:

1. W widoku Właściwości produktu Integration Designer wybierz opcję **Powiązanie > Propagacja**.
2. Ustaw wymagane opcje propagowania nagłówka transportu.

Uwaga: Propagowanie nagłówków transportu jest domyślnie wyłączone i może być wdrożone tylko w środowisku wykonawczym w wersji 7.0.0.3 (lub późniejszej wersji). Należy również zauważyć, że w przypadku wersji 7.0.0.3 propagowanie nagłówków transportu jest możliwe tylko w przypadku nagłówków transportu HTTP.

Jeśli zostanie włączone propagowanie nagłówków transportu, nagłówki będą propagowane z komunikatów odebranych przez moduł i jeśli nie zostaną one jawnie usunięte, będą używane w kolejnych wywołaniach w tym samym wątku.

Uwaga: Nagłówków transportu nie można propagować, jeśli używane jest powiązanie usługi Web Service JAX-RPC.

Uzyskiwanie dostępu do informacji w nagłówku

Jeśli włączone jest propagowanie nagłówków transportu dla odbieranych komunikatów, wszystkie nagłówki transportu (w tym nagłówki niestandardowe) są widoczne w obiekcie komunikatu usługi (SMO). Nagłówkom można nadać inne wartości, ale można również utworzyć nowe nagłówki. Należy jednak zauważyć, że nie jest przeprowadzane żadne sprawdzanie ani sprawdzanie poprawności ustawianych wartości. Niepoprawne lub nieodpowiednie nagłówki mogą zaś powodować problemy z usługą Web Service w czasie wykonywania.

Podczas ustawiania nagłówków HTTP należy uwzględnić następujące informacje:

- Wszystkie zmiany nagłówków, które są zastrzeżone dla mechanizmu usługi Web Service, nie będą uwzględniane w komunikacie wychodzącym. Nagłówkami zastrzeżonymi dla mechanizmu usługi Web Service są na przykład: wersja HTTP, metoda HTTP, nagłówek Content-Type, nagłówek Content-Length i nagłówek SOAPAction.
- Jeśli wartość nagłówka jest liczbą, należy ustawić bezpośrednio liczbę (a nie łańcuch). Na przykład należy użyć definicji **Max-Forwards = 5** (a nie **Max-Forwards = Max-Forwards: 5**) lub **Age = 300** (a nie **Age = Age: 300**).
- Jeśli komunikat żądania jest mniejszy niż 32 kB, mechanizm usługi Web Service usuwa nagłówek Transfer-Encoding i ustawia nagłówek Content-Length na ustaloną wielkość komunikatu.
- Nagłówek Content-language jest resetowany przez mechanizm WAS.channel.http w ścieżce odpowiedzi.
- Niepoprawne ustawienie nagłówka Upgrade powoduje błąd 500.
- Następujące nagłówki dopisują wartość zastrzeżoną przez mechanizm usługi Web Service do ustawień klienta:
 - User-Agent,
 - Cache-Control,
 - Pragma,
 - Accept,
 - Connection.

Dostęp do informacji w nagłówku można uzyskać na jeden z następujących sposobów:

- Korzystając z operacji podstawowej mediacji w celu uzyskania dostępu do struktur obiektu SMO
Informacje na temat używania operacji podstawowych mediacji zawiera sekcja z odsyłaczami “Informacje pokrewne”.
- Korzystając z interfejsu SPI usługi kontekstu

Poniższy przykładowy kod odczytuje nagłówki transportu HTTP z usługi kontekstu:

```
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();
    }
}
```

Rozwiązywanie problemów

Jeśli podczas wysyłania zmienionych nagłówków wystąpią problemy, można przechwycić komunikat TCP/IP za pomocą takich narzędzi jak monitor TCP/IP w produkcie Integration Designer. Aby uzyskać dostęp do monitora TCP/IP, należy wybrać opcję **Uruchom/debuguj > Monitor TCP/IP** na stronie Preferencje.

Wartości nagłówków można również wyświetlić przy użyciu śledzenia mechanizmu JAX-WS: **org.apache.axis2.*=all: com.ibm.ws.websvcs.*=all:**

Praca z powiązaniem usługi Web Services JAX-WS: **Advanced**

Jeśli w aplikacjach używane są powiązania usługi Web Service JAX-WS, istnieje możliwość dodania jakości usług (QoS) SAML (Security Assertion Markup Language) do powiązania. W tym celu należy najpierw użyć Konsoli administracyjnej do zaimportowania zestawu strategii. Konsoli administracyjnej można również użyć, aby upewnić się, czy serwer jest poprawnie skonfigurowany pod kątem użycia wraz z powiązaniem usługi Web Service JAX-WS.

Importowanie zestawów strategii SAML: **Advanced**

Język Security Assertion Markup Language (SAML) jest opartym na formacie XML standardem OASIS umożliwiającym wymianę informacji o tożsamości użytkownika i atrybutach zabezpieczeń. Zestaw strategii SAML można określić, konfigurując powiązanie usługi Web Service JAX-WS w produkcie Integration Designer. Najpierw za pomocą Konsoli administracyjnej produktu IBM Business Process Manager należy udostępnić zestawy strategii SAML, aby można je było zaimportować do produktu Integration Designer.

Zestawy strategii SAML zwykle znajdują się w katalogu konfiguracji profilu:

katalog_główny_profilu/config/templates/PolicySets

Przed rozpoczęciem wykonywania tej procedury należy sprawdzić, czy następujące katalogi (zawierające zestawy strategii) znajdują się w katalogu konfiguracji profilu:

- SAML11 Bearer WSHTTPS default
- SAML20 Bearer WSHTTPS default
- SAML11 Bearer WSSecurity default
- SAML20 Bearer WSSecurity default
- SAML11 HoK Public WSSecurity default
- SAML20 HoK Public WSSecurity default
- SAML11 HoK Symmetric WSSecurity default
- SAML20 HoK Symmetric WSSecurity default

- Username WSHTTPS default

Jeśli katalogów tych nie ma w katalogu konfiguracji profilu, należy je tam skopiować z następującego położenia:

katalog_główny_serwera_aplikacji/profileTemplates/default/documents/config/templates/PolicySets

Importując zestawy strategii do Konsoli administracyjnej, należy wybrać te z nich, które mają być dostępne dla produktu Integration Designer, a następnie zapisać plik .zip dla każdego z tych zestawów strategii w miejscu dostępnym dla produktu Integration Designer.

1. Zaimportuj zestawy strategii, wykonując następujące kroki:
 - a. W Konsoli administracyjnej wybierz opcję **Usługi > Zestawy strategii > Zestawy strategii aplikacji**.
 - b. Kliknij opcję **Importuj > Z repozytorium domyślnego**.
 - c. Wybierz domyślne zestawy strategii SAML i kliknij przycisk **OK**.
2. Wyeksportuj zestawy strategii, aby można było ich użyć w produkcie Integration Designer:
 - a. Na stronie zestawów strategii aplikacji wybierz zestaw strategii SAML, który należy wyeksportować, i kliknij opcję **Eksportuj**.

Uwaga: Jeśli strona zestawów strategii aplikacji nie jest aktualnie wyświetlana, kliknij opcję **Usługi > Zestawy strategii > Zestawy strategii aplikacji** w Konsoli administracyjnej.

- b. Na następnej stronie kliknij odsyłacz do pliku .zip zestawu strategii.
- c. W oknie Pobieranie pliku kliknij przycisk **Zapisz** i wskaż miejsce dostępne dla produktu Integration Designer.
- d. Kliknij przycisk **Wstecz**.
- e. Wykonaj kroki od 2a na stronie 78 do 2d na stronie 78 dla każdego eksportowanego zestawu strategii.

Zestawy strategii SAML zostaną zapisane w plikach .zip i będą gotowe do zaimportowania do produktu Integration Designer.

Zestawy strategii należy zaimportować do produktu Integration Designer w sposób opisany w temacie “Zestawy strategii”.

Wywoływanie usług Web Services wymagających podstawowego uwierzytelniania HTTP: **Advanced**

W podstawowym uwierzytelnianiu HTTP do uwierzytelnienia klienta usługi wobec zabezpieczonego punktu końcowego stosowane są nazwa użytkownika oraz hasło. Podstawowe uwierzytelnianie HTTP można skonfigurować na potrzeby wysyłania i odbierania żądań usług Web Services.

Skonfigurowanie podstawowego uwierzytelniania HTTP na potrzeby odbierania żądań usługi Web Service polega na skonfigurowaniu interfejsu API języka Java API dla powiązania elementu eksportowanego XML usług Web Services JAX-WS w sposób opisany w temacie Tworzenie i przypisywanie ról zabezpieczeń do elementów eksportowanych usług Web Service.

Podstawowe uwierzytelnianie HTTP można włączyć na potrzeby żądań usług Web Services, które są wysyłane przez powiązanie elementu importowanego JAX-WS, w jeden z dwóch sposobów:

- Podczas konfigurowania powiązania elementu importowanego w module SCA można wybrać dostarczony zestaw strategii uwierzytelniania HTTP o nazwie BPMHTTPBasicAuthentication (udostępniany wraz z powiązaniem elementu importowanego usługi Web Service JAX-WS) lub dowolny inny zestaw strategii obejmujący strategię HTTPTransport.
- Podczas konstruowania modułu SCA można użyć możliwości przepływu mediacji do dynamicznego utworzenia nowego nagłówka uwierzytelniania HTTP oraz określenia w tym nagłówku informacji o nazwie użytkownika i hasle.

Uwaga: Zestaw strategii ma pierwszeństwo przed wartością określoną w nagłówku. Aby w czasie wykonywania użyć wartości ustawionej w nagłówku uwierzytelniania HTTP, nie należy załączać zestawu strategii obejmującego strategię

HTTPTransport. W szczególności nie należy używać domyślnego zestawu strategii BPMHTTPBasicAuthentication oraz, jeśli zdefiniowano zestaw strategii, należy upewnić się, że nie obejmuje on strategii HTTPTransport.

Więcej informacji na temat zestawów strategii i powiązań strategii usług Web Services, jak również sposobu ich używania można znaleźć w temacie Web services policy sets (Zestawy strategii usług Web Services) w Centrum informacyjnym serwera WebSphere Application Server.

- Aby użyć dostarczonego zestawu strategii, wykonaj następujące kroki:
 1. Opcjonalne: W Konsoli administracyjnej utwórz powiązanie strategii ogólnej klienta obejmującej strategię HTTPTransport z wartościami wymaganego identyfikatora użytkownika i hasła lub przeprowadź edycję istniejącego.
 2. W produkcie IBM Integration Designer wygeneruj powiązanie elementu importowanego usługi Web Service JAX-WS i przyłącz do niego zestaw strategii BPMHTTPBasicAuthentication.
 3. Wykonaj *jeden* z następujących kroków:
 - W produkcie IBM Integration Designer we właściwościach powiązania elementu importowanego usługi Web Service JAX-WS określ nazwę istniejącego powiązania strategii ogólnej klienta obejmującej strategię HTTPTransport.
 - Po wdrożeniu modułu SCA użyj Konsoli administracyjnej, aby wybrać istniejące powiązanie strategii klienta lub utworzyć nowe i powiązać je z powiązaniem importu.
 4. Opcjonalne: W Konsoli administracyjnej serwera Process Server przeprowadź edycję wybranego powiązania zestawu strategii, określając wymagany identyfikator i hasło.
- Aby określić nazwę użytkownika i hasło w nagłówku uwierzytelniania HTTP, wykonaj jeden z następujących zestawów kroków:
 - Użyj w produkcie IBM Integration Designer operacji podstawowej mediacji procedury ustawiającej nagłówek HTTP do utworzenia nagłówka uwierzytelniania HTTP i określ nazwę użytkownika oraz hasło.
 - Jeśli wymagana jest dodatkowa logika, użyj kodu w języku Java w niestandardowej operacji podstawowej mediacji (w sposób przedstawiony w poniższym przykładzie) realizującej następujące działania:
 1. Utworzenie nagłówka uwierzytelniania HTTP.
 2. Określenie nazwy użytkownika i hasła.
 3. Dodanie nowego nagłówka uwierzytelniania HTTP do elementu sterującego HTTPControl.
 4. Ustawienie zaktualizowanego elementu sterującego HTTPControl z powrotem w usłudze Kontekst.

```
// Pobranie typu HeaderInfoType z usługi kontekstu
ContextService contextService = (ContextService) ServiceManager.INSTANCE
    .locateService("com/ibm/bpm/context/ContextService");
HeaderInfoType headers = contextService.getHeaderInfo();
if(headers == null){
    headers = ContextObjectFactory.eINSTANCE.createHeaderInfoType();
}
// Pobranie nagłówka HTTP i elementu sterującego HTTPControl z typu HeaderInfoType
HTTPHeaderType httpHeaderType = headers.getHTTPHeader();
HTTPControl cp = httpHeaderType.getControl();
HeadersFactory factory = HeadersFactory.eINSTANCE;
if(cp == null){
    cp = factory.createHTTPControl();
}
// Utworzenie nowego obiektu HTTPAuthentication i ustawienie referencji 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);
// Ustawienie informacji nagłówka z powrotem na bieżący kontekst wykonywania.
contextService.setHeaderInfo(headers);
```

Sprawdzanie konfiguracji serwera: **Advanced**

Wdrażając aplikację z powiązaniem z usługą Web Service JAX-WS, należy upewnić się, że serwer, na którym ta aplikacja jest wdrożona, nie ma zaznaczonej opcji **Start components as needed** (Uruchamiaj komponenty, gdy są potrzebne).

Aby sprawdzić, czy opcja ta jest zaznaczona, wykonaj następujące kroki w Konsoli administracyjnej:

1. Kliknij opcję **Serwery > Typy serwerów > Serwery aplikacji WebSphere**.
2. Kliknij nazwę serwera.
3. Na karcie Konfiguracja sprawdź, czy opcja **Start components as needed** (Uruchamiaj komponenty, gdy są potrzebne) jest zaznaczona.
4. Wykonaj jeden z następujących kroków:
 - Jeśli opcja **Start components as needed** (Uruchamiaj komponenty, gdy są potrzebne) jest zaznaczona, usuń zaznaczenie i kliknij przycisk **Zastosuj**.
 - Jeśli opcja **Start components as needed** (Uruchamiaj komponenty, gdy są potrzebne) nie jest zaznaczona, kliknij przycisk **Anuluj**.

Załączniki w komunikatach SOAP: **Advanced**

Istnieje możliwość wysyłania i odbierania komunikatów SOAP, które zawierają dane binarne (na przykład pliki PDF lub obrazy JPEG) dołączone w formie załączników. Załączniki mogą być *przywoływane* (czyli reprezentowane jawnie jako części komunikatu w interfejsie usługi) lub *nieprzywoływane* (w tym przypadku możliwe jest dołączenia dowolnej liczby i dowolnego typu załączników).

Przywoływany załącznik może być reprezentowany w jeden z następujących sposobów:

- W załącznikach MTOM stosowane jest kodowanie używane w mechanizmie SOAP Message Transmission Optimization Mechanism (<http://www.w3.org/TR/soap12-mtom/>). Możliwość korzystania z załączników MTOM włącza się za pomocą odpowiedniej opcji konfiguracyjnej w powiązaniach importu i eksportu. Jest to zalecana metoda kodowania załączników dla nowych aplikacji.
- Jako element wsi:swaRef-typed w schemacie komunikatu
Załączniki zdefiniowane przy użyciu typu wsi:swaRef są zgodne ze standardem organizacji WS-I (Web Services Interoperability Organization) *Attachments Profile Version 1.0* (profil załącznika, wersja 1.0 - <http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), który definiuje relacje między elementami komunikatu i częściami MIME.
- Jako część komunikatu najwyższego poziomu, przy użyciu typu schematu binarnego
Załączniki reprezentowane jako części komunikatu najwyższego poziomu są zgodne ze specyfikacją *SOAP Messages with Attachments* (komunikaty SOAP z załącznikami - <http://www.w3.org/TR/SOAP-attachments>).
Załączniki reprezentowane jako części komunikatu najwyższego poziomu mogą również być skonfigurowane tak, aby zagwarantować że dokument WSDL i komunikaty utworzone przez powiązanie są zgodne ze standardami WS-I *Attachments Profile Version 1.0* (profil załącznika, wersja 1.0) i WS-I *Basic Profile Version 1.1* (profil podstawowy, wersja 1.1 - <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>).

Nieprzywoływany załącznik jest przenoszony w komunikacie SOAP bez żadnej reprezentacji w schemacie komunikatu.

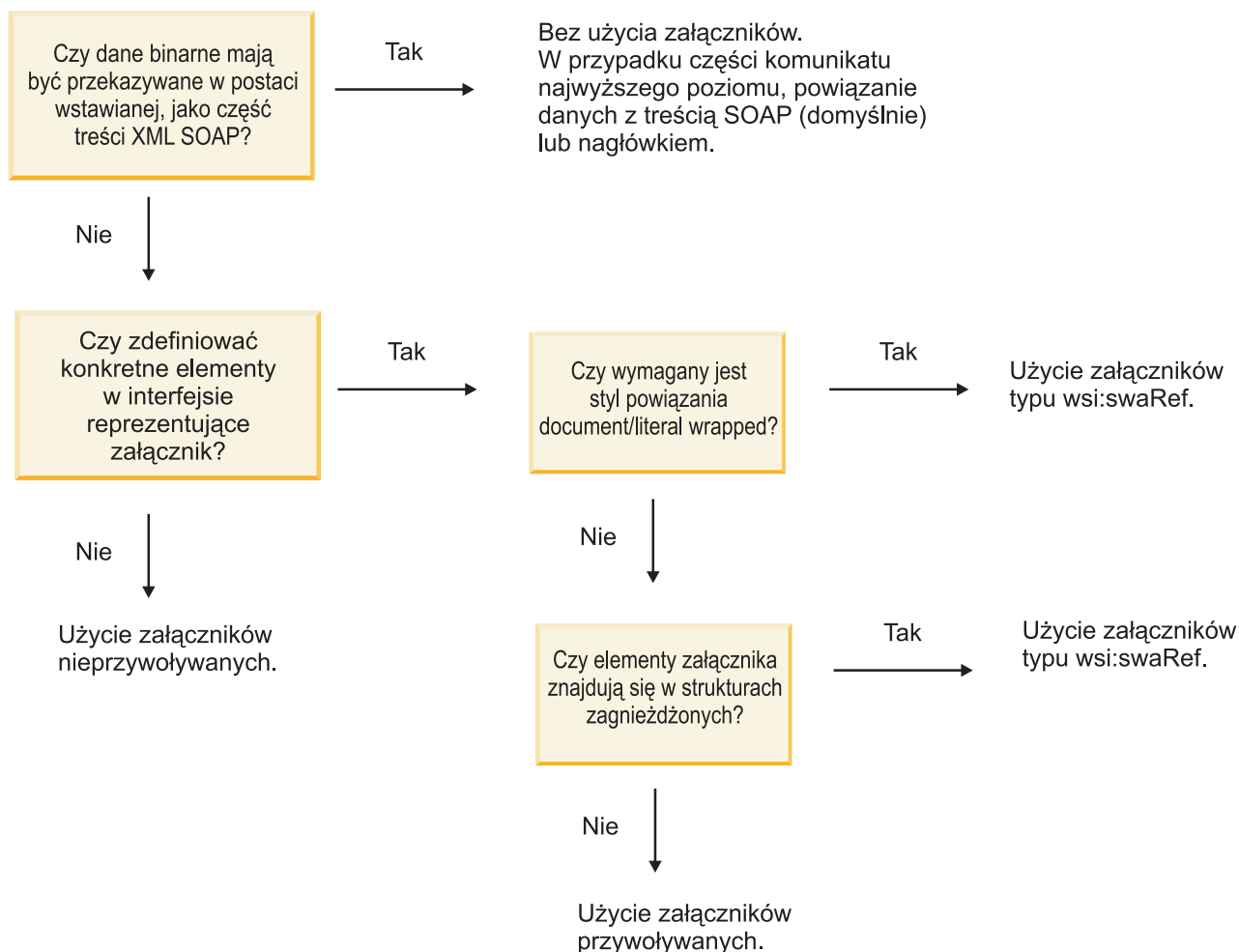
We wszystkich przypadkach, z wyjątkiem załączników MTOM, powiązanie SOAP WSDL powinno zawierać powiązanie MIME dla załączników, które mają zostać użyte, a maksymalna wielkość załączników nie powinna przekraczać 20 MB.

Uwaga: Aby wysyłać lub odbierać komunikaty SOAP z załącznikami, należy używać jednego z powiązań usług Web Services opartego na specyfikacji JAX-WS (Java API for XML Web Services).

Jak wybrać odpowiedni styl załącznika: **Advanced**

Podczas projektowania nowego interfejsu usługi, który zawiera dane binarne, należy uwzględnić sposób przenoszenia tych danych binarnych w komunikatach SOAP wysyłanych i odbieranych przez usługę.

Mechanizm Message Transmission Optimization Mechanism (MTOM) należy stosować na potrzeby załączników, gdy obsługuje go podłączona aplikacja usługi Web Service. Jeśli tak nie jest, należy wybrać inny styl załączników. Jest to przedstawione na poniższym diagramie. Przy określaniu odpowiedniego stylu załącznika należy posłużyć się następującymi pytaniami:



Załączniki MTOM: części komunikatu najwyższego poziomu:

Istnieje możliwość wysyłania i odbierania komunikatów usług Web Services zawierających załączniki MTOM (Message Transmission Optimization Mechanism) SOAP. W komunikacie SOAP z wieloma częściami MIME pierwszą częścią komunikatu jest treść SOAP, a załącznik lub załączniki znajdują się w kolejnych częściach.

Przy wysyłaniu lub odbieraniu przywoływanego załącznika w komunikacie SOAP dane binarne, które tworzą załącznik (często dość duży), są przechowywane niezależnie od treści komunikatu SOAP, a więc nie muszą być analizowane jako kod XML. Powoduje to, że przetwarzanie jest wydajniejsze niż byłoby, gdyby dane binarne były przechowywane w elemencie XML.

Poniżej przedstawiony jest przykładowy komunikat MTOM SOAP:

```

... inne nagłówki transportowe ...
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>

... w tym miejscu są dane binarne ...
--MIMEBoundaryurn_uuid_0FE43E4D025F0BF3DC11582467646812--

```

Należy zwrócić uwagę, że w tym przykładzie mechanizmu MTOM typem treści koperty SOAP jest **application/xop+xml**, a dane binarne są zastąpione następującym elementem **xop:Include**:

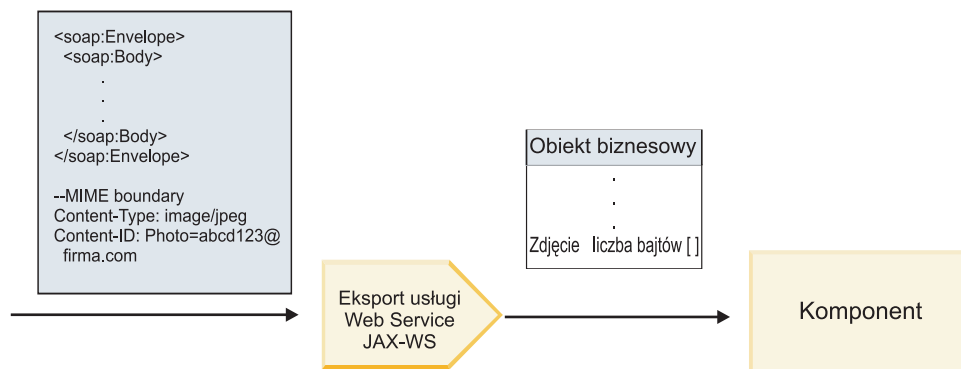
```

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

```

Przetwarzanie danych przychodzących załączników przywoływanych

Jeśli klient przekaże komunikat SOAP z załącznikiem do komponentu SCA (Service Component Architecture), powiązanie eksportu usługi Web Service JAX-WS najpierw usuwa załącznik. Następnie analizuje część SOAP komunikatu i tworzy obiekt biznesowy. Na koniec, powiązanie umieszcza binarny załącznik w obiekcie biznesowym.



Rysunek 58. Sposób przetwarzania przez powiązanie eksportu usługi Web Service (JAX-WS) komunikatu SOAP z załącznikiem przywoływanym

Atrybuty załączników MTOM

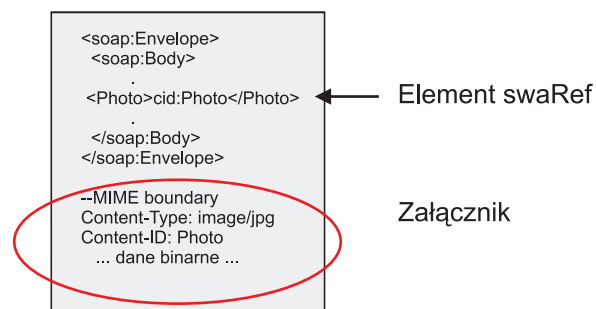
- Mechanizm MTOM może obsługiwać elementy załącznika w obrębie struktur zagnieżdżonych.
- Mechanizm MTOM jest dostępny jedynie dla typu base64Binary.
- Mechanizm MTOM może obsługiwać elementy załącznika w obrębie struktur zagnieżdżonych. Oznacza to, że elementy **bodyPath** związane z załącznikami MTOM określają położenie **xpath** elementu, w którym przechowywany jest załącznik MTOM. Logika obliczania wartości elementu **bodyPath** ściśle odpowiada schematowi generowania położenia **xpath** przedstawionemu w następujących przykładach:
 - W przypadku typu innego niż tablicowy (**maxOccurs** jest równe 1): /sendImage/input/imageData
 - W przypadku typu tablicowego (**maxOccurs** > 1): /sendImage/input/imageData[1]
- Mieszane typy załączników nie są obsługiwane. Oznacza to, że jeśli mechanizm MTOM jest włączony w powiązaniu importu, generowany będzie załącznik MTOM. Jeśli mechanizm MTOM jest wyłączony lub jeśli pozostawiono domyślną wartość konfiguracji mechanizmu MTOM w powiązaniu eksportu, przychodzący komunikat MTOM nie jest obsługiwany.

Załączniki przywoływane: elementy typu swaRef: **Advanced**

Istnieje możliwość wysyłania i odbierania komunikatów SOAP zawierających załączniki reprezentowane w interfejsie usługi jako elementy typu swaRef.

Element typu swaRef jest zdefiniowany w standardzie organizacji WS-I (Web Services Interoperability Organization) *Attachments Profile* Version 1.0 (profil załącznika, wersja 1.0 - <http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), który definiuje relacje między elementami komunikatu i częściami MIME.

Treść komunikatu SOAP zawiera element typu swaRef określający identyfikator treści załącznika.



Rysunek 59. Komunikat SOAP z elementem swaRef

Dokument WSDL dla tego komunikatu SOAP zawiera element swaRef w części komunikatu identyfikującej załącznik.

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

Dokument WSDL powinien też zawierać powiązanie MIME wskazujące, że mają być używane wieloczęściowe komunikaty MIME.

Uwaga: Dokument WSDL *nie* zawiera powiązania MIME dla konkretnego elementu komunikatu typu swaRef, ponieważ powiązania MIME dotyczą tylko części komunikatu najwyższego poziomu.

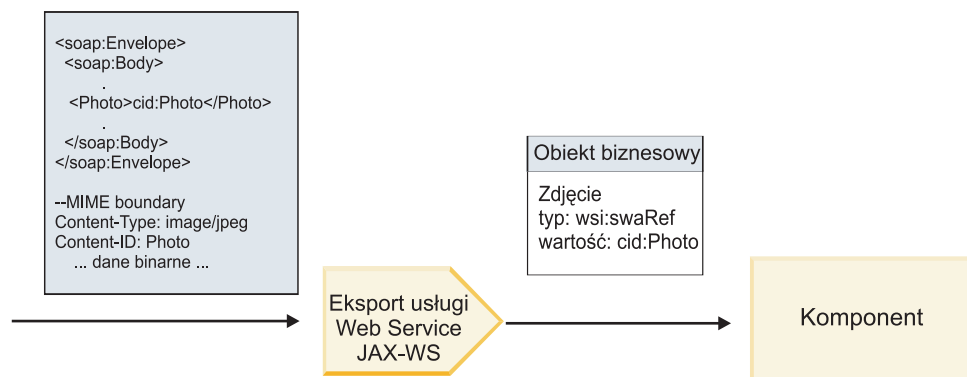
Załączniki reprezentowane jako elementy typu swaRef mogą być propagowane tylko przez komponenty przepływu mediacji. Jeśli załącznik wymaga dostępu z poziomu innego typu komponentu lub propagowania do innego typu komponentu, należy użyć komponentu przepływu mediacji w celu przeniesienia załącznika do miejsca, które jest dostępne dla tego komponentu.

Przetwarzanie danych przychodzących załączników

Do konfigurowania powiązania eksportu w celu odebrania załącznika używany jest produkt Integration Designer. Użytkownik tworzy moduł i powiązany interfejs oraz operacje, między innymi element typu swaRef. Następnie tworzy powiązanie usługi Web Service JAX-WS.

Uwaga: Szczegółowe informacje zawiera temat “Praca z załącznikami” w Centrum informacyjnym produktu Integration Designer.

Jeśli klient przekaże komunikat SOAP z załącznikiem swaRef do komponentu SCA (Service Component Architecture), powiązanie eksportu usługi Web Service JAX-WS najpierw usuwa załącznik. Następnie analizuje część SOAP komunikatu i tworzy obiekt biznesowy. Na koniec, powiązanie ustawia identyfikator treści załącznika w obiekcie biznesowym.



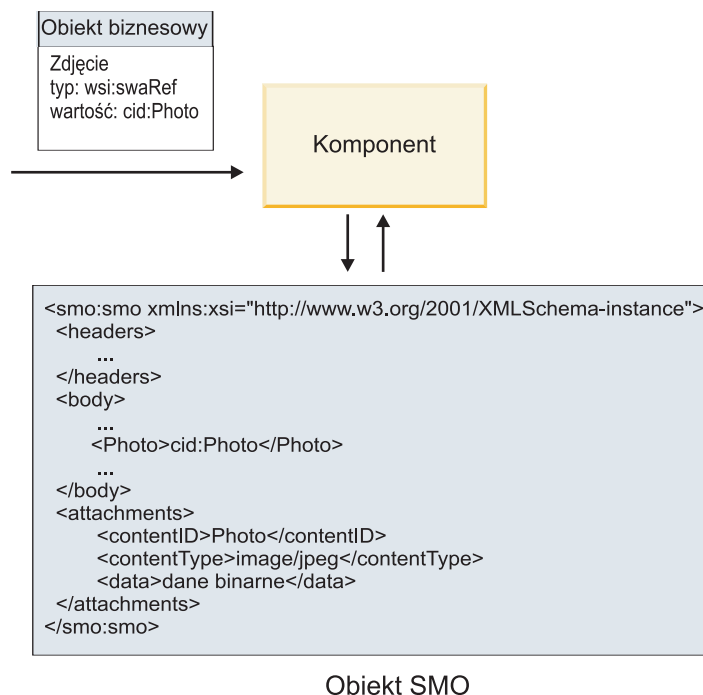
Rysunek 60. Sposób przetwarzania komunikatu SOAP zawierającego załącznik swaRef przez powiązanie eksportu usługi Web Service JAX-WS

Uzyskiwanie dostępu do metadanych załącznika w komponencie przepływu mediacji

Jak przedstawiono na ilustracji Rys. 16 na stronie 85, kiedy komponenty uzyskują dostęp do załączników swaRef, identyfikator treści załącznika jest wyświetlany jako element typu swaRef.

Dla każdego załącznika komunikatu SOAP istnieje odpowiadający mu element **attachments** w obiekcie SMO. Jeśli używany jest typ swaRef profilu WS-I, element **attachments** zawiera typ treści załącznika oraz identyfikator treści, a także rzeczywiste dane binarne załącznika.

Aby uzyskać wartość załącznika swaRef, konieczne jest więc uzyskanie wartości elementu typu swaRef, a następnie znalezienie elementu **attachments** z odpowiednią wartością **contentID**. Należy zauważyć, że w przypadku wartości **contentID** przedrostek **cid:** jest zazwyczaj usuwany z wartości swaRef.



Rysunek 61. Sposób wyświetlania załączników swaRef w obiekcie SMO

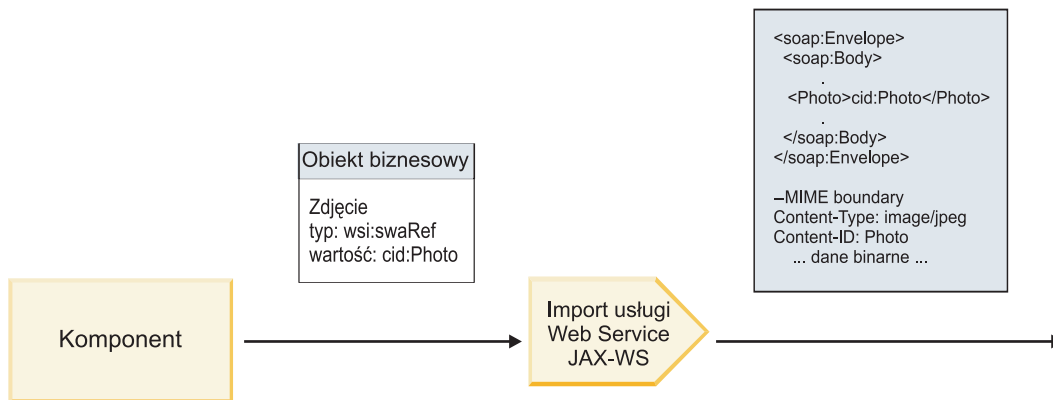
Przetwarzanie danych wychodzących

Do skonfigurowania powiązania importu usługi Web Service JAX-WS w celu wywołania zewnętrznej usługi Web Service używany jest produkt Integration Designer. Powiązanie importu jest konfigurowane przy użyciu dokumentu WSDL, który opisuje usługę Web Service do wywołania oraz definiuje załącznik, który zostanie przekazany do usługi Web Service.

Po otrzymaniu komunikatu SCA przez powiązanie importu usługi Web Service JAX-WS, elementy typu swaRef są wysyłane jako załączniki, jeśli import jest połączony z komponentem przepływu mediacji, a element typu swaRef ma odpowiedni element **attachments**.

W przypadku przetwarzania danych wychodzących elementy typu swaRef są zawsze wysyłane wraz z ich wartościami identyfikatora treści. Jednak moduł mediacji musi gwarantować, że istnieje odpowiedni element **attachments** ze zgodną wartością **contentID**.

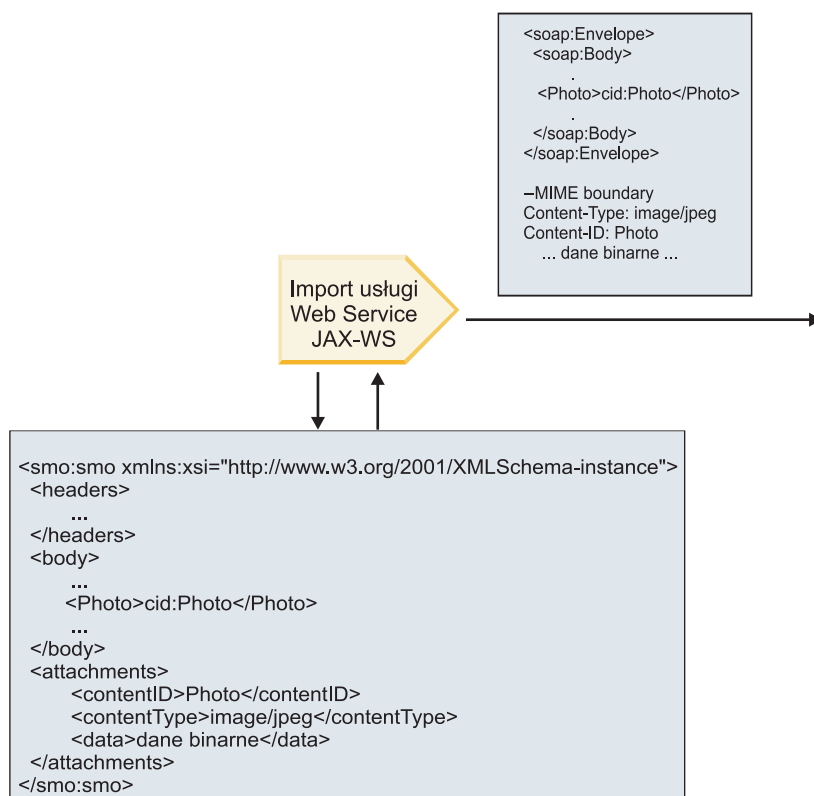
Uwaga: Aby zachować zgodność z profilem załączników WS-I, wartość **content ID** powinna być zgodna z kodowaniem części identyfikatora content-id opisanym w sekcji 3.8 dokumentu WS-I *Attachments Profile* 1.0.



Rysunek 62. Sposób generowania komunikatu SOAP zawierającego załącznik swaRef przez powiązanie importu usługi Web Service JAX-WS

Ustawianie metadanych załącznika w komponencie przepływu mediacji

Jeśli w obiekcie SMO występuje wartość elementu typu swaRef oraz element **attachments**, powiązanie przygotowuje komunikat SOAP (wraz z załącznikiem) i wysyła ten komunikat do odbiorcy.



Obiekt SMO

Rysunek 63. Sposób dostępu do załącznika swaRef w obiekcie SMO w celu utworzenia komunikatu SOAP

Obiekt SMO zawiera element **attachments** tylko jeśli komponent przepływu mediacji jest połączony bezpośrednio z importem lub eksportem. Element ten nie jest przekazywany przez inne typy komponentów. Jeśli potrzebne są wartości znajdujące się w module zawierającym inne typy komponentów, należy użyć komponentu przepływu mediacji w celu skopiowania tych wartości do miejsca, w którym będzie możliwe uzyskiwanie do nich dostępu z poziomu modułu.

Należy również użyć innego komponentu przepływu mediacji do ustawienia poprawnych wartości przed zrealizowaniem wywołania wychodzącego za pośrednictwem importu usługi Web Service.

Ważne: Operacja podstawowa mediacji Odzworowanie transformuje komunikaty przy użyciu transformacji XSLT 1.0. Zostało to opisane w sekcji “Reprezentacja XML obiektu SMO”. Transformacja opiera się na serializacji XML obiektu SMO. Operacja podstawowa mediacji Odzworowanie umożliwia określenie elementu głównej serializacji, a element główny dokumentu XML odzwierciedla ten element głównej serializacji.

Podczas wysyłania komunikatów SOAP z załącznikami wybrany element główny określa sposób propagowania załączników.

- Jeśli jako element główny odzworowania XML zostanie użyty element “/body”, wszystkie załączniki są domyślnie propagowane do całego odzworowania.
- Jeśli jako element główny odzworowania zostanie użyty element “/”, można kontrolować propagowanie załączników.

Załączniki przywoływane: części komunikatu najwyższego poziomu: 

Istnieje możliwość wysyłania i odbierania komunikatów SOAP zawierających załączniki binarne, które są zadeklarowane jako części w interfejsie usługi.

W komunikacie SOAP z wieloma częściami MIME pierwszą częścią komunikatu jest treść SOAP, a załącznik lub załączniki znajdują się w kolejnych częściach.

Jaka korzyść płynie z wysyłania lub odbierania przywoływanego załącznika w komunikacie SOAP? Dane binarne, które tworzą załącznik (często dość duży) są przechowywane niezależnie od treści komunikatu SOAP, a więc nie muszą być analizowane jako kod XML. Powoduje to, że przetwarzanie jest wydajniejsze niż byłoby, gdyby dane binarne były przechowywane w elemencie XML.

Typy komunikatów SOAP z załącznikami przywoływanymi

Od wersji 7.0.0.3 produktu IBM Business Process Manager użytkownik ma wybór sposobu generowania komunikatu SOAP:

- **Komunikaty zgodne ze standardem WS-I**

Środowisko wykonawcze może generować komunikaty SOAP, które są zgodne ze standardem WS-I *Attachments Profile Version 1.0* (profil załącznika, wersja 1.0) i standardem WS-I *Basic Profile Version 1.1* (profil podstawowy, wersja 1.1). W komunikacie SOAP zgodnym z tymi standardami tylko jedna część komunikatu jest powiązana z treścią komunikatu SOAP. W przypadku tych części, które są powiązane jako załączniki, do powiązania załącznika z częścią komunikatu używane jest kodowanie części identyfikatora content-id (opisane w standardzie WS-I *Attachments Profile Version 1.0*).

- **Komunikaty niezgodne ze standardami WS-I**

Środowisko wykonawcze może generować komunikaty SOAP, które nie są zgodne ze standardami WS-I, ale są zgodne z komunikatami generowanymi w wersji 7.0 lub 7.0.0.2 produktu IBM Business Process Manager. Komunikaty SOAP używają elementów najwyższego poziomu o nazwach utworzonych na podstawie części komunikatu, które zawierają atrybut **href** przechowujący identyfikator **content-id** załącznika, ale nie jest używane kodowanie części identyfikatora content-id (opisane w standardzie WS-I *Attachments Profile Version 1.0*).

Wybór zgodności ze standardami WS-I w przypadku eksportów usług Web Services

Do konfigurowania powiązania eksportu używany jest produkt Integration Designer. Użytkownik tworzy moduł i powiązany interfejs oraz operacje. Następnie tworzy powiązanie usługi Web Service JAX-WS. Strona Załączniki przywoływane wyświetla wszystkie binarne części z utworzonej operacji, a użytkownik wybiera części, które będą załącznikami. Następnie na stronie Określanie zgodności ze standardem WS-I AP 1.0 produktu Integration Designer określana jest jedna z następujących opcji:

- **Użyj komunikatu SOAP zgodnego ze standardem WS-I AP 1.0**

Jeśli zostanie wybrana ta opcja, użytkownik określa również która część komunikatu powinna być powiązana z treścią komunikatu SOAP.

Uwaga: Tej opcji można użyć tylko w sytuacji, kiedy odpowiedni plik WSDL jest również zgodny ze standardem WS-I.

Plik WSDL wygenerowany przez produkt Integration Designer 7.0.0.3 jest zgodny ze standardem WS-I. Jeśli jednak zostanie zaimportowany plik WSDL, który nie jest zgodny ze standardem WS-I, nie można wybrać tej opcji.

- **Użyj komunikatu SOAP niezgodnego ze standardem WS-I AP 1.0**

Jeśli zostanie wybrana ta opcja (która jest opcją domyślną), z treścią komunikatu SOAP powiązana jest pierwsza część komunikatu.

Uwaga: Jako załączniki przywoływane można wysyłać tylko części komunikatu najwyższego poziomu (czyli elementy zdefiniowane w elemencie portType pliku WSDL jako części komunikatu wejściowego lub wyjściowego), które mają typ binarny (base64Binary lub hexBinary).

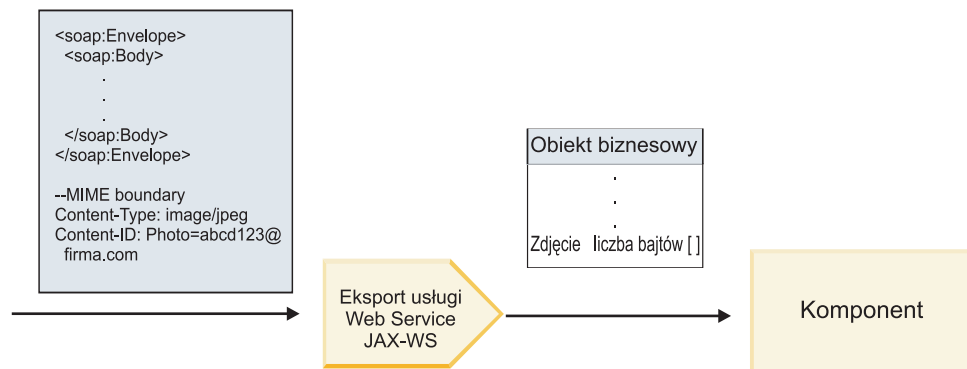
Szczegółowe informacje zawiera temat “Praca z załącznikami” w Centrum informacyjnym produktu Integration Designer.

W przypadku komunikatów zgodnych ze standardem WS-I wygenerowany w komunikacie SOAP identyfikator content-ID jest konkatencją następujących elementów:

- wartości atrybutu **name** elementu **wSDL:part** przywoływanego przez element **mime:content**,
- znaku **=**,
- globalnie unikalnej wartości, na przykład identyfikatora UUID,
- znaku **@**,
- poprawnej nazwy domeny.

Przetwarzanie danych przychodzących załączników przywoływanych

Jeśli klient przekaże komunikat SOAP z załącznikiem do komponentu SCA (Service Component Architecture), powiązanie eksportu usługi Web Service JAX-WS najpierw usuwa załącznik. Następnie analizuje część SOAP komunikatu i tworzy obiekt biznesowy. Na koniec, powiązanie umieszcza binarny załącznik w obiekcie biznesowym.



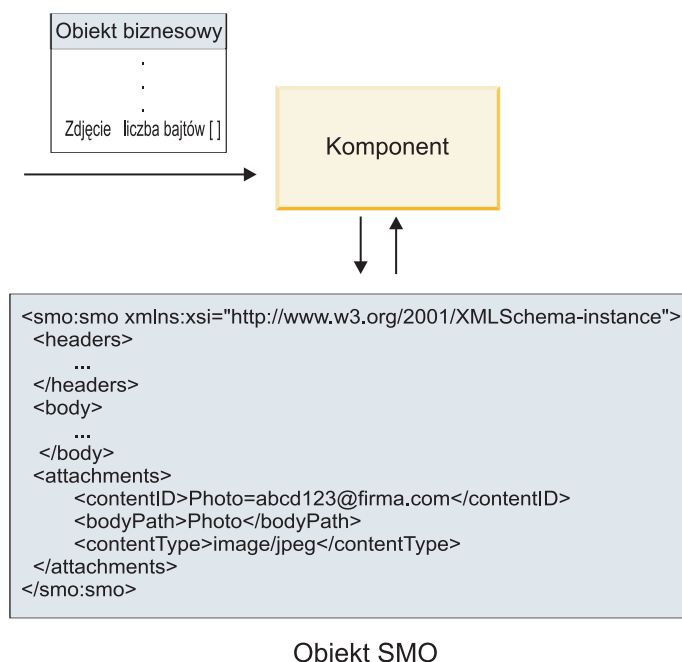
Rysunek 64. Sposób przetwarzania komunikatu SOAP zgodnego ze standardem WS-I i zawierającego załącznik przywoływany przez powiązanie eksportu usługi Web Service JAX-WS

Uzyskiwanie dostępu do metadanych załącznika w komponencie przepływu mediacji

Jak przedstawiono na ilustracji Rys. 19 na stronie 88, kiedy komponenty uzyskują dostęp do załączników przywoływanych, dane załącznika są wyświetlane jako tablica bajtów.

Dla każdego załącznika przywoływanego komunikatu SOAP istnieje odpowiadający mu element **attachments** w obiekcie SMO. Element **attachments** zawiera typ treści załącznika oraz ścieżkę do elementu treści komunikatu, w

którym przechowywany jest załącznik.



Rysunek 65. Sposób wyświetlania załączników przywoływanych w obiekcie SMO

Ważne: Ścieżka do elementu treści komunikatu nie jest automatycznie aktualizowana, jeśli komunikat zostanie poddany transformowaniu, a załącznik zostanie przeniesiony. Istnieje możliwość użycia przepływu mediacji do zaktualizowania elementu **attachments** przy użyciu nowej ścieżki (na przykład w ramach transformowania lub za pomocą oddzielnej procedury ustawiającej element komunikatu).

Sposób konstruowania wychodzących komunikatów SOAP

Do skonfigurowania powiązania importu usługi Web Service JAX-WS w celu wywołania zewnętrznej usługi Web Service używany jest produkt Integration Designer. Powiązanie importu jest skonfigurowane przy użyciu dokumentu WSDL, który opisuje usługę Web Service do wywołania oraz definiuje części komunikatu, które mają być przekazywane jako załączniki. Na stronie Określanie zgodności ze standardem WS-I AP 1.0 produktu Integration Designer można również określić jedną z następujących opcji:

- **Użyj komunikatu SOAP zgodnego ze standardem WS-I AP 1.0**

Jeśli zostanie wybrana ta opcja, użytkownik określa również która część komunikatu powinna być powiązana z treścią komunikatu SOAP. Wszystkie pozostałe części są powiązane z załącznikami lub nagłówkami. Komunikaty wysyłane przez powiązanie nie obejmują elementów w treści komunikatu SOAP, które odwołują się do załączników. Relacja jest wyrażana za pośrednictwem identyfikatora treści załącznika zawierającego nazwę części komunikatu.

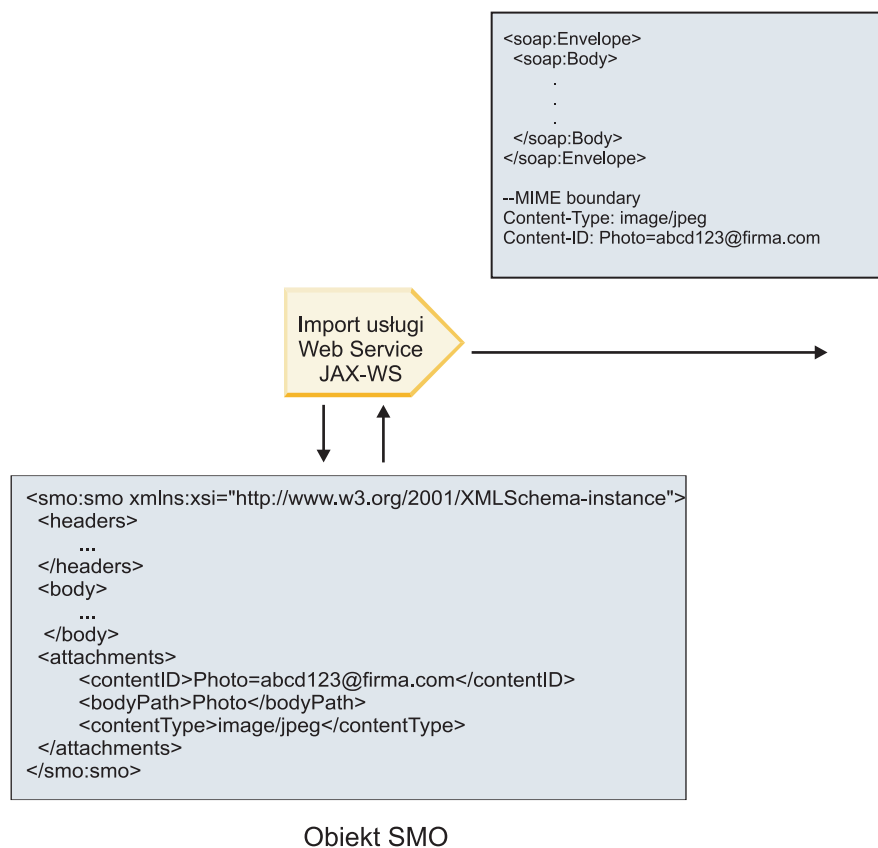
- **Użyj komunikatu SOAP niezgodnego ze standardem WS-I AP 1.0**

Jeśli zostanie wybrana ta opcja (która jest opcją domyślną), z treścią komunikatu SOAP powiązana jest pierwsza część komunikatu. Wszystkie pozostałe części są powiązane z załącznikami lub nagłówkami. Komunikaty wysyłane przez powiązanie zawierają w treści komunikatu SOAP co najmniej jeden element odwołujący się do załączników przy użyciu atrybutu **href**.

Uwaga: Część reprezentująca załącznik, zdefiniowana w dokumencie WSDL, musi być typem prostym (base64Binary lub hexBinary). Jeśli część jest zdefiniowana przy użyciu typu złożonego (complexType), ta część nie może być powiązana jako załącznik.

Przetwarzanie danych wychodzących załączników przywoływanych

Powiązanie importu korzysta z informacji w obiekcie SMO do określania sposobu, w jaki binarne części komunikatu najwyższego poziomu są wysyłane w postaci załączników.



Rysunek 66. Sposób dostępu do załącznika przywoływanego w obiekcie SMO w celu utworzenia komunikatu SOAP

Obiekt SMO zawiera element **attachments** tylko jeśli komponent przepływu mediacji jest połączony bezpośrednio z importem lub eksportem. Element ten nie jest przekazywany przez inne typy komponentów. Jeśli potrzebne są wartości znajdujące się w module zawierającym inne typy komponentów, należy użyć komponentu przepływu mediacji w celu skopiowania tych wartości do miejsca, w którym będzie możliwe uzyskiwanie do nich dostępu z poziomu modułu. Należy również użyć innego komponentu przepływu mediacji do ustawienia poprawnych wartości przed zrealizowaniem wywołania wychodzącego za pośrednictwem importu usługi Web Service.

Powiązanie określa sposób wysyłania komunikatu (lub określa, czy komunikat ma zostać wysłany) na podstawie kombinacji następujących warunków:

- Czy istnieje powiązanie WSDL MIME dla binarnej części komunikatu najwyższego poziomu, a jeśli tak, w jaki sposób zdefiniowany jest typ treści.
- Czy w obiekcie SMO istnieje element **attachments**, którego wartość **bodyPath** odwołuje się do binarnej części najwyższego poziomu.

Sposób tworzenia załączników, jeśli w obiekcie SMO istnieje element attachment

Poniższa tabela przedstawia sposób tworzenia i wysyłania załącznika, jeśli obiekt SMO zawiera element **attachment** z wartością **bodyPath** zgodną z częścią nazwy komunikatu:

Tabela 61. Sposób generowania załącznika

Status powiązania WSDL MIME dla binarnej części komunikatu najwyższego poziomu	Sposób tworzenia i wysyłania komunikatu
<p>Jest obecne, z jednym z następujących elementów:</p> <ul style="list-style-type: none"> • brakiem zdefiniowanego typu treści dla części komunikatu, • wieloma zdefiniowanymi typami treści, • typem treści zdefiniowanym przy użyciu znaku wieloznacznego. 	<p>Część komunikatu jest wysyłana jako załącznik.</p> <p>Identyfikator Content-Id jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest generowany.</p> <p>Parametr Content-Type jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest ustawiany na wartość application/octet-stream.</p>
<p>Jest obecne, z pojedynczą treścią (bez znaku wieloznacznego) dla części komunikatu</p>	<p>Część komunikatu jest wysyłana jako załącznik.</p> <p>Identyfikator Content-Id jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest generowany.</p> <p>Parametr Content-Type jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest ustawiany na typ zdefiniowany w elemencie treści WSDL MIME.</p>
<p>Brak</p>	<p>Część komunikatu jest wysyłana jako załącznik.</p> <p>Identyfikator Content-Id jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest generowany.</p> <p>Parametr Content-Type jest ustawiany na wartość z elementu attachments (jeśli ten element istnieje). W przeciwnym razie jest ustawiany na wartość application/octet-stream.</p> <p>Uwaga: Wysyłanie części komunikatów w postaci załączników, jeśli nie są one tak zdefiniowane w dokumencie WSDL, może spowodować utracenie zgodności ze standardem WS-I Attachments Profile 1.0. Należy więc tego unikać, jeśli to możliwe.</p>

Sposób tworzenia załączników, jeśli w obiekcie SMO nie istnieje element attachment

Poniższa tabela przedstawia sposób tworzenia i wysyłania załącznika, jeśli obiekt SMO nie zawiera elementu **attachment** z wartością **bodyPath** zgodną z częścią nazwy komunikatu:

Tabela 62. Sposób generowania załącznika

Status powiązania WSDL MIME dla binarnej części komunikatu najwyższego poziomu	Sposób tworzenia i wysyłania komunikatu
<p>Jest obecne, z jednym z następujących elementów:</p> <ul style="list-style-type: none"> • brakiem zdefiniowanego typu treści dla części komunikatu, • wieloma zdefiniowanymi typami treści, • typem treści zdefiniowanym przy użyciu znaku wieloznacznego. 	<p>Część komunikatu jest wysyłana jako załącznik.</p> <p>Identyfikator Content-Id jest generowany.</p> <p>Parametr Content-Type jest ustawiany na wartość application/octet-stream.</p>
<p>Jest obecne, z pojedynczą treścią (bez znaku wieloznacznego) dla części komunikatu</p>	<p>Część komunikatu jest wysyłana jako załącznik.</p> <p>Identyfikator Content-Id jest generowany.</p> <p>Parametr Content-Type jest ustawiany na typ zdefiniowany w elemencie treści WSDL MIME.</p>

Tabela 62. Sposób generowania załącznika (kontynuacja)

Status powiązania WSDL MIME dla binarnej części komunikatu najwyższego poziomu	Sposób tworzenia i wysyłania komunikatu
Brak	Część komunikatu nie jest wysyłana jako załącznik.

Ważne: Operacja podstawowa mediacji Odwzorowanie transformuje komunikaty przy użyciu transformacji XSLT 1.0. Zostało to opisane w sekcji “Reprezentacja XML obiektu SMO”. Transformacja opiera się na serializacji XML obiektu SMO. Operacja podstawowa mediacji Odwzorowanie umożliwia określenie elementu głównego serializacji, a element główny dokumentu XML odzwierciedla ten element główny serializacji.

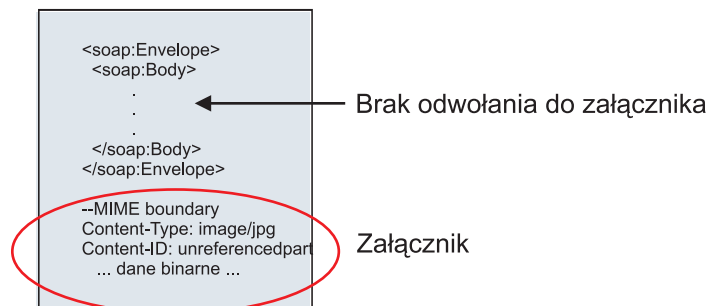
Podczas wysyłania komunikatów SOAP z załącznikami wybrany element główny określa sposób propagowania załączników.

- Jeśli jako element główny odwzorowania XML zostanie użyty element “/body”, wszystkie załączniki są domyślnie propagowane do całego odwzorowania.
- Jeśli jako element główny odwzorowania zostanie użyty element “/”, można kontrolować propagowanie załączników.

Załączniki nieprzywoływane: **Advanced**

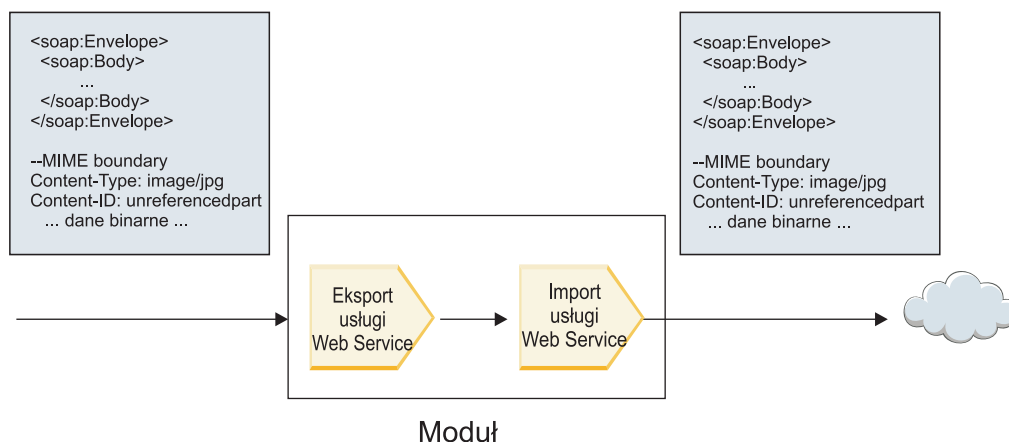
Istnieje możliwość wysyłania i odbierania załączników *nieprzywoływanych*, które nie są deklarowane w interfejsie usługi.

W komunikacie SOAP z wieloma częściami MIME pierwszą częścią komunikatu jest treść SOAP, a załączniki znajdują się w kolejnych częściach. W treści SOAP nie występuje żadne odwołanie do załącznika.



Rysunek 67. Komunikat SOAP z załącznikiem nieprzywoływanym

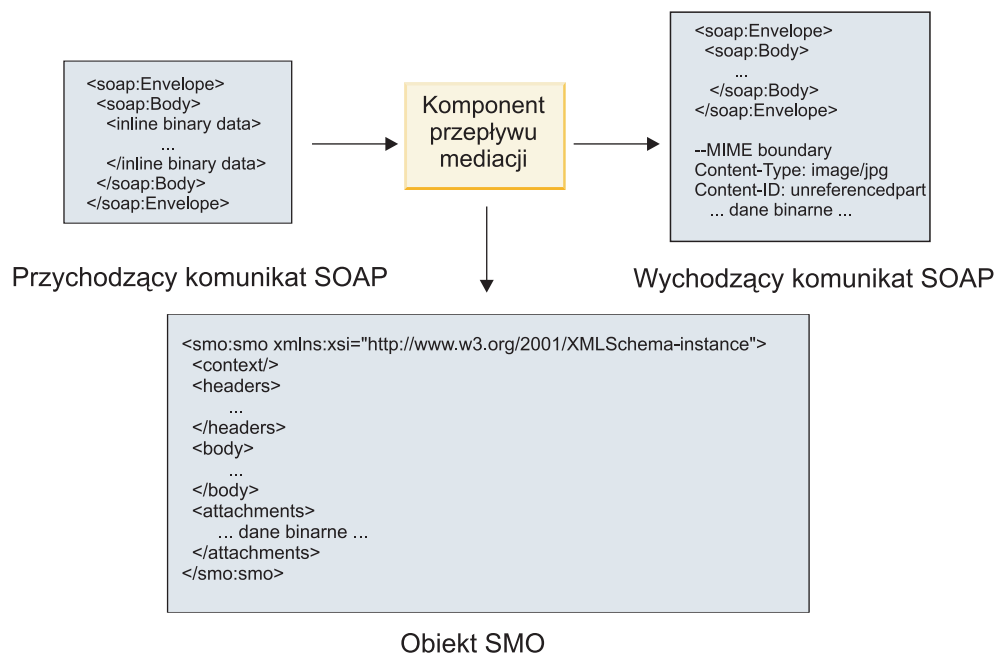
Komunikat SOAP z załącznikiem nieprzywoływanym można wysłać za pośrednictwem eksportu usługi Web Service do importu usługi Web Service. Załącznik jest zawarty w komunikacie wyjściowym wysyłanym do docelowej usługi Web Service.



Rysunek 68. Załącznik przechodzący przez moduł SCA

Na ilustracji Rys. 23 na stronie 93 przedstawiono komunikat SOAP z załącznikiem, który jest przekazywany bez modyfikowania.

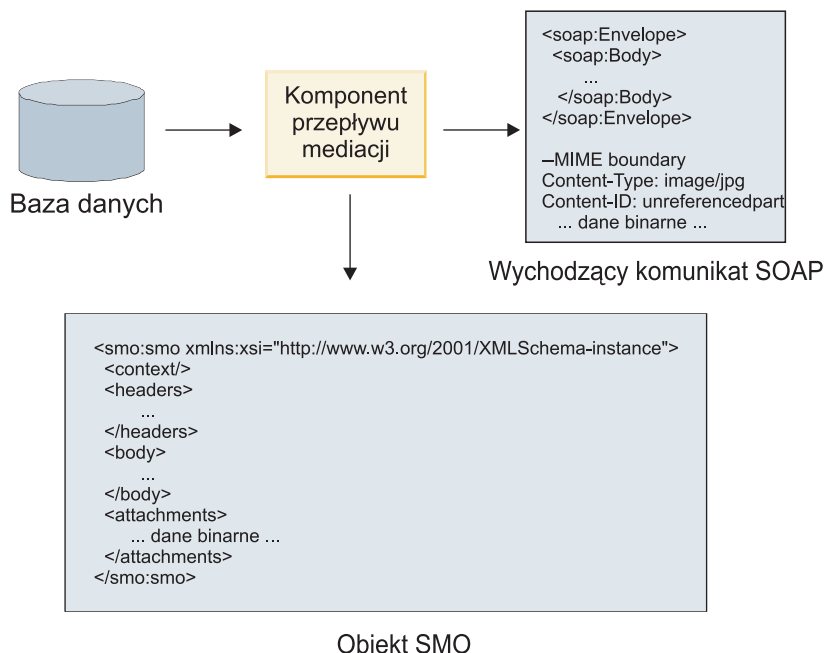
Istnieje również możliwość zmodyfikowania komunikatu SOAP przy użyciu komponentu przepływu mediacji. Można na przykład użyć komponentu przepływu mediacji do wyodrębnienia danych z komunikatu SOAP (w tym przypadku - danych binarnych zawartych w treści komunikatu) i utworzenia komunikatu SOAP z załącznikami. Dane są przetwarzane w ramach elementu załączników obiektu SMO.



Rysunek 69. Komunikat przetworzony przez komponent przepływu mediacji

Komponent przepływu mediacji może również transformować komunikat przychodzący, wyodrębniając i kodując załącznik, a następnie przesyłając komunikat bez załączników.

Zamiast wyodrębniania danych z przychodzącego komunikatu SOAP w celu utworzenia komunikatu SOAP z załącznikami, można uzyskać dane załącznika ze źródła zewnętrznego, na przykład bazy danych.



Rysunek 70. Załącznik uzyskany z bazy danych i dodany do komunikatu SOAP

Komponent przepływu mediacji może również wyodrębnić załącznik z przychodzącego komunikatu SOAP i przetworzyć ten komunikat (na przykład zapisując załącznik w bazie danych).

Załączniki nieprzywoływane mogą być propagowane tylko przez komponenty przepływu mediacji. Jeśli załącznik wymaga dostępu z poziomu innego typu komponentu lub propagowania do innego typu komponentu, należy użyć komponentu przepływu mediacji w celu przeniesienia załącznika do miejsca, które jest dostępne dla tego komponentu.

Ważne: Operacja podstawowa mediacji Odzworowanie transformuje komunikaty przy użyciu transformacji XSLT 1.0. Zostało to opisane w sekcji "Reprezentacja XML obiektu SMO". Transformacja opiera się na serializacji XML obiektu SMO. Operacja podstawowa mediacji Odzworowanie umożliwia określenie elementu głównego serializacji, a element główny dokumentu XML odzwierciedla ten element główny serializacji.

Podczas wysyłania komunikatów SOAP z załącznikami wybrany element główny określa sposób propagowania załączników.

- Jeśli jako element główny odwzorowania XML zostanie użyty element `</body>`, wszystkie załączniki są domyślnie propagowane do całego odwzorowania.
- Jeśli jako element główny odwzorowania zostanie użyty element `</>`, można kontrolować propagowanie załączników.

Używanie powiązania stylu dokumentu WSDL z komunikatami wieloczęściowymi: **Advanced**

Organizacja WS-I (Web Services Interoperability Organization) zdefiniowała zbiór reguł sposobu opisywania usług Web Services za pomocą dokumentu WSDL oraz sposobu formułowania odpowiednich komunikatów SOAP w celu zagwarantowania współdziałania.

Te reguły są określone w standardzie *WS-I Basic Profile Version 1.1* (profil podstawowy, wersja 1.1 - <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>). W szczególności dokument WS-I Basic Profile 1.1 R2712 określa, że: „Powiązanie w stylu document-literal MUSI być serializowane jako element ENVELOPE z elementem soap:Body, którego element potomny jest instancją deklaracji elementu globalnego przywoływanej przez odpowiednią część `wSDL:message`”.

Oznacza to, że podczas używania powiązania SOAP w stylu document w operacji z komunikatami (wejściowymi, wyjściowymi lub błędu) zdefiniowanymi z wieloma częściami, tylko jedna z tych części powinna być powiązana z treścią komunikatu SOAP, aby zachować zgodność z profilem WS-I Basic Profile 1.1.

Dodatkowo dokument WS-I Attachments Profile 1.0 R2941 określa, że: „Element wsdl:binding w elemencie DESCRIPTION POWINIEN wiązać każdy element wsdl:part elementu wsdl:message w elemencie wsdl:portType, do którego się odwołuje, z jednym z elementów: soapbind:body, soapbind:header, soapbind:fault, soapbind:headerfault lub mime:content”.

Oznacza to, że podczas używania powiązania SOAP w stylu document w operacji z komunikatami (wejściowymi, wyjściowymi lub błędu) zdefiniowanymi z wieloma częściami, wszystkie części inne niż ta, która została wybrana do powiązania z treścią komunikatu SOAP, powinny być powiązane jako załączniki lub nagłówki.

Podczas generowania opisów WSDL na potrzeby eksportów z powiązaniem usług Web Services (JAX-WS i JAX-RPC) jest w tym przypadku stosowane następujące podejście:

- Użytkownik może wybrać, która część komunikatu jest powiązana z treścią komunikatu SOAP, jeśli istnieje więcej niż jeden element typu innego niż binarny. Jeśli istnieje tylko jeden element typu innego niż binarny, ten element jest automatycznie powiązany z treścią komunikatu SOAP.
- W przypadku powiązania JAX-WS wszystkie inne części komunikatu typu hexBinary lub base64Binary są powiązane jako załączniki przywoływane. Patrz temat “Załączniki przywoływane: części komunikatu najwyższego poziomu” na stronie 87.
- Wszystkie inne części komunikatu są powiązane jako nagłówki SOAP.

Powiązania importu JAX-RPC i JAX-WS uwzględniają powiązanie SOAP w istniejącym dokumencie WSDL zawierającym komunikaty w stylu multipart document, nawet jeśli nie wiąże on wielu części z treścią komunikatu SOAP. Nie ma jednak możliwości wygenerowania klientów usług Web Services dla takich dokumentów WSDL w produkcie Rational Application Developer.

Uwaga: Powiązanie JAX-RPC nie obsługuje załączników.

Zalecany wzorec podczas używania komunikatów wieloczęściowych z operacją, która ma powiązanie SOAP w stylu document jest więc następujące podejście:

1. Używanie stylu document/literal wrapped. W tym przypadku komunikaty zawsze mają jedną część, ale załączniki muszą być nieprzywoływane (jak opisano w temacie “Załączniki nieprzywoływane” na stronie 92) lub muszą mieć typ swaRef (jak opisano w temacie “Załączniki przywoływane: elementy typu swaRef” na stronie 83).
2. Używanie stylu RPC/literal. W tym przypadku nie ma ograniczeń powiązania WSDL w zakresie liczby części powiązanych z treścią komunikatu SOAP. Wynikowe komunikaty SOAP zawsze mają pojedynczy element potomny reprezentujący wywoływaną operację, a części komunikatu są elementami potomnymi tego elementu.
3. W przypadku powiązania JAX-WS - stosowanie maksymalnie jednej części komunikatu, która nie ma typu hexBinary ani base64Binary, chyba że dopuszczalne jest powiązanie pozostałych części innych niż binarne z nagłówkami SOAP.
4. Wszystkie inne przypadki podlegają opisanemu sposobowi działania.

Uwaga: Istnieją dodatkowe ograniczenia używania komunikatów SOAP niezgodnych z profilem WS-I *Basic Profile Version 1.1*.

- Pierwsza część komunikatu powinna być inna niż binarna.
- Podczas odbierania komunikatów SOAP w stylu multipart document z załącznikami przywoływanymi powiązanie JAX-WS oczekuje, że każdy załącznik przywoływany będzie reprezentowany przez element potomny treści komunikatu SOAP z wartością atrybutu href identyfikującą załącznik na podstawie jego identyfikatora treści. Powiązanie JAX-WS wysyła załączniki przywoływane dla takich komunikatów, wykorzystując ten sam sposób. To działanie nie jest zgodne z podstawowym profilem WS-I.

Aby zagwarantować, że komunikaty są zgodne z profilem podstawowym, należy postępować zgodnie z punktem 1 na stronie 95 lub punktem 2 na stronie 95 na wcześniej liście lub unikać używania załączników przywoływanych dla takich komunikatów i zamiast nich użyć załączników nieprzywoływanych lub załączników typu swaRef.

Powiązania HTTP

Powiązanie HTTP zostało zaprojektowane, aby udostępniać architekturze SCA (Service Component Architecture) połączenia z protokołem HTTP. Dzięki niemu istniejące lub nowo utworzone aplikacje HTTP mogą uczestniczyć w środowiskach architektury zorientowanej na usługi (SOA).

Protokół HTTP (Hypertext Transfer Protocol) jest powszechnie używanym protokołem służącym do przesyłania informacji w sieci WWW. Jeśli użytkownik pracuje z zewnętrzną aplikacją używającą protokołu HTTP, niezbędne jest zastosowanie powiązania HTTP. Powiązanie HTTP transformuje dane przekazywane jako komunikat w formacie rodzimym do obiektu biznesowego w aplikacji SCA. Powiązanie HTTP może również transformować dane przekazywane jako obiekt biznesowy na format rodzimy oczekiwany przez aplikację zewnętrzną.

Uwaga: Aby prowadzić interakcję z klientami i usługami korzystającymi z protokołu SOAP/HTTP usług Web Services, należy rozważyć użycie jednego z powiązań usługi Web Service. Powiązania te udostępniają dodatkową funkcjonalność, umożliwiającą obsługiwanie standardowych elementów jakości usług w usługach Web Services.

Niektóre typowe scenariusze używania powiązania HTTP zostały wymienione na poniższej liście:

- Usługi udostępniane przez architekturę SCA mogą wywoływać aplikacje HTTP przy użyciu importu HTTP.
- Usługi udostępniane przez architekturę SCA mogą ujawniać się jako aplikacje z funkcjami HTTP, aby mogły być używane przez klienty HTTP za pośrednictwem eksportu HTTP.
- IBM Business Process Manager i Process Server mogą komunikować się ze sobą za pośrednictwem infrastruktury HTTP, a więc użytkownicy mogą zarządzać komunikacją przy zastosowaniu standardów korporacyjnych.
- IBM Business Process Manager i Process Server mogą działać jako mediatory komunikacji HTTP, transformując i kierując komunikaty, co usprawnia integrację aplikacji przy użyciu sieci HTTP.
- IBM Business Process Manager i Process Server mogą być używane do utworzenia mostu między protokołem HTTP i innymi protokołami takimi jak protokół SOAP/HTTP usług Web Services, adaptery oparte na zasobach JCA (Java Connector Architecture), protokół JMS itd.

Szczegółowe informacje na temat tworzenia powiązań importu i eksportu HTTP zawiera Centrum informacyjne produktu Integration Designer. Należy zapoznać się z tematami **Tworzenie aplikacji integracji > Uzyskiwanie dostępu do usług zewnętrznych przy użyciu protokołu HTTP**.

Przegląd powiązań HTTP:

Powiązanie HTTP udostępnia połączenia z aplikacjami udostępnianymi za pośrednictwem protokołu HTTP. Mediuje ono komunikację między aplikacjami HTTP i umożliwia wywoływanie istniejących aplikacji opartych na protokole HTTP z modułu.

Powiązania importu HTTP

Powiązanie importu HTTP udostępnia połączenia wychodzące z aplikacji SCA (Service Component Architecture) do serwera HTTP lub aplikacji HTTP.

Import wywołuje adres URL punktu końcowego protokołu HTTP. Adres URL można określić na jeden z trzech sposobów:

- Adres URL może być ustawiany dynamicznie w nagłówkach HTTP za pośrednictwem dynamicznie nadpisywanego adresu URL.
- Adres URL może być ustawiany dynamicznie w elemencie docelowego adresu obiektu SMO.
- Adres URL może być określany jako właściwość konfiguracji w imporcie.

To wywołanie jest zawsze synchroniczne.

Mimo że wywołania HTTP są zawsze wywołaniami typu żądanie-odpowiedź, import HTTP obsługuje zarówno operacje jednokierunkowe, jak i dwukierunkowe oraz ignoruje odpowiedzi w przypadku operacji jednokierunkowych.

Powiązania eksportu HTTP

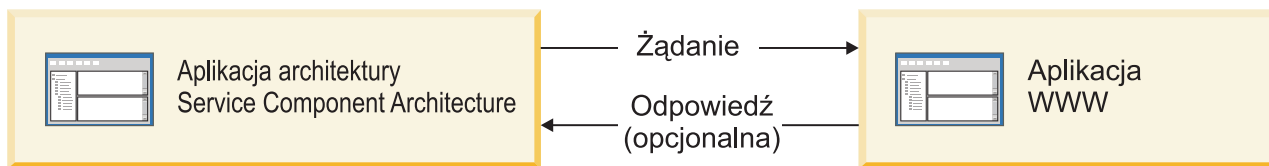
Powiązanie eksportu HTTP udostępnia połączenia przychodzące z aplikacji HTTP do aplikacji SCA.

Adres URL jest definiowany w eksporcie HTTP. Aplikacje HTTP, które wymagają wysyłania komunikatów żądania do eksportu, używają tego adresu URL do wywołania eksportu.

Eksport HTTP obsługuje również komendy ping.

Powiązania HTTP w czasie wykonywania

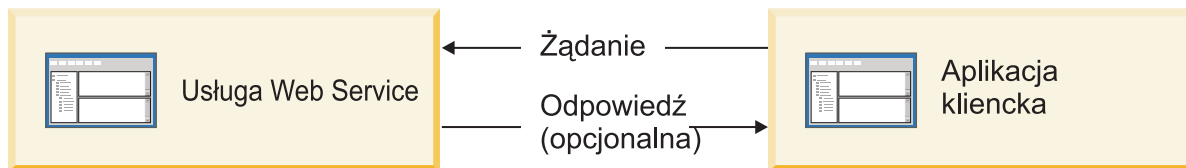
W czasie wykonywania import z powiązaniem HTTP wysyła żądanie z aplikacji SCA do zewnętrznej usługi Web Service, zawierające dane w treści komunikatu lub niezawierające tych danych. Żądanie jest wykonywane z aplikacji SCA do zewnętrznej usługi Web Service. Zostało to przedstawione na następującej ilustracji: Rys. 26 na stronie 97.



Rysunek 71. Przepływ żądania z aplikacji SCA do aplikacji WWW

Opcjonalnie import z powiązaniem HTTP może odbierać dane z aplikacji WWW (w odpowiedzi na żądanie).

W przypadku eksportu żądanie jest wykonywane przez aplikację kliencką do usługi Web Service. Zostało to przedstawione na następującej ilustracji: Rys. 27 na stronie 97.



Rysunek 72. Przepływ żądania z aplikacji klienckiej do usługi Web Service

Usługa Web Service jest aplikacją WWW działającą na serwerze. Eksport jest implementowany w tej aplikacji WWW w postaci serwletu, a więc klient wysyła żądanie do adresu URL. Serwlet przekazuje żądanie do aplikacji SCA w środowisku wykonawczym.

Opcjonalnie eksport może wysyłać dane do aplikacji klienckiej (w odpowiedzi na żądanie).

Nagłówki HTTP:

Powiązania importu i eksportu HTTP pozwalają, aby konfiguracja nagłówków HTTP oraz ich wartości były używane na potrzeby komunikatów wychodzących. Import HTTP używa tych nagłówków na potrzeby żądań, a eksport HTTP używa ich na potrzeby odpowiedzi.

Statycznie skonfigurowane nagłówki i informacje kontrolne są nadrzędne w stosunku do wartości ustawianych dynamicznie w czasie wykonywania. Jednak dynamicznie nadpisywane wartości kontrolne adresu URL, wersji i metody przesłaniają wartości statyczne, które są w przeciwnym razie uznawane za domyślne.

Powiązanie obsługuje dynamiczny rodzaj adresu URL importu HTTP, określając wartość docelowego adresu URL protokołu HTTP, wersję i metodę w czasie wykonywania. Te wartości są określane poprzez wyodrębnianie wartości odwołania punktu końcowego, dynamicznie nadpisywanego adresu URL, wersji oraz metody.

- W przypadku odwołania punktu końcowego należy użyć funkcji API `com.ibm.websphere.sca.addressing.EndpointReference` lub ustawić pole `/headers/SMOHeader/Target/address` w nagłówku obiektu SMO.
- W przypadku dynamicznie nadpisywanego adresu URL, wersji i metody, należy użyć sekcji parametrów sterujących protokołu HTTP w komunikacie SCA (Service Component Architecture). Należy zauważyć, że dynamicznie nadpisywany adres URL ma pierwszeństwo w kolejności wykonywania operacji przed docelowym odwołaniem punktu końcowego. Z kolei odwołanie punktu końcowego ma zastosowanie w wielu powiązaniach, a więc jest uznawane za preferowane podejście i należy go używać tam, gdzie to możliwe.

Informacje kontrolne i informacje w nagłówku dla komunikatów wychodzących w powiązaniach eksportu i importu HTTP są przetwarzane w następującej kolejności:

1. Informacje w nagłówku i informacje kontrolne z wyjątkiem dynamicznie nadpisywanego adresu URL protokołu HTTP, wersji i metody z komunikatu SCA (najniższy priorytet).
2. Zmiany dokonane w Konsoli administracyjnej na poziomie eksportu/importu.
3. Zmiany dokonane w Konsoli administracyjnej na poziomie metody eksportu lub importu.
4. Adres docelowy określony za pośrednictwem odwołania punktu końcowego lub nagłówka obiektu SMO.
5. Dynamicznie nadpisywany adres URL, wersja i metoda z komunikatu SCA.
6. Informacje w nagłówku i informacje kontrolne z procedury obsługi danych lub powiązania danych (najwyższy priorytet).

Eksport i import HTTP wypełniają nagłówki wychodzące i parametry sterujące danymi z komunikatu przychodzącego (HTTPExportRequest i HTTPImportResponse) tylko, jeśli propagacja nagłówków protokołów jest ustawiona na wartość **True**. Eksport i import HTTP odczytują i przetwarzają nagłówki wychodzące i parametry sterujące (HTTPExportResponse i HTTPImportRequest) tylko, jeśli propagacja nagłówków protokołów jest ustawiona na wartość **True**.

Uwaga: Zmiany wprowadzone przez procedury obsługi danych lub powiązania danych do nagłówków lub parametrów sterujących w odpowiedzi importu lub żądaniu eksportu nie będą miały wpływu na instrukcje przetwarzania komunikatu wewnątrz powiązania importu lub eksportu i powinny być używane tylko w celu propagowania zmodyfikowanych wartości do zagnieżdżonych komponentów SCA.

W ramach usługi kontekstu jest wykonywane propagowanie kontekstu (w tym nagłówków protokołów, takich jak nagłówek HTTP, i kontekstu użytkownika, na przykład identyfikatora konta) w ścieżce wywołań SCA. Podczas programowania w produkcie IBM Integration Designer użytkownik może kontrolować propagowanie kontekstu za pośrednictwem właściwości importu i eksportu. Więcej szczegółów można znaleźć w informacjach na temat powiązań importów i eksportów w Centrum informacyjnym produktu IBM Integration Designer.

Dostarczane struktury nagłówka HTTP i obsługa

Tabela Tabela 34 na stronie 98 przedstawia parametry żądania/odpowiedzi dla żądań i odpowiedzi importu HTTP i eksportu HTTP.

Tabela 63. Dostarczane informacje nagłówka HTTP

Nazwa elementu sterującego	Żądanie importu HTTP	Odpowiedź importu HTTP	Żądanie eksportu HTTP	Odpowiedź eksportu HTTP
URL (adres URL)	Zignorowano	Nieustawiony	Odczytywany z komunikatu żądania. Uwaga: Łańcuch zapytania również stanowi część parametru sterującego adresu URL.	Zignorowano

Tabela 63. Dostarczane informacje nagłówka HTTP (kontynuacja)

Nazwa elementu sterującego	Żądanie importu HTTP	Odpowiedź importu HTTP	Żądanie eksportu HTTP	Odpowiedź eksportu HTTP
Version (wersja; możliwe wartości: 1.0, 1.1; wartość domyślna to 1.1)	Zignorowano	Nieustawiony	Odczytywany z komunikatu żądania	Zignorowano
Method (metoda)	Zignorowano	Nieustawiony	Odczytywany z komunikatu żądania	Zignorowano
Dynamic Override URL (dynamicznie nadpisywany adres URL)	Jeśli jest ustawiony w procedurze obsługi danych lub powiązaniu danych, nadpisuje adres URL importu HTTP. Zapisywany w komunikacie w wierszu żądania. Uwaga: Łańcuch zapytania również stanowi część parametru sterującego adresu URL.	Nieustawiony	Nieustawiony	Zignorowano
Dynamic Override Version (dynamicznie nadpisywana wersja)	Jeśli jest ustawiony, nadpisuje wersję importu HTTP. Zapisywany w komunikacie w wierszu żądania.	Nieustawiony	Nieustawiony	Zignorowano
Dynamic Override Method (dynamicznie nadpisywana metoda)	Jeśli jest ustawiony, nadpisuje metodę importu HTTP. Zapisywany w komunikacie w wierszu żądania.	Nieustawiony	Nieustawiony	Zignorowano
Media Type (typ nośnika; ten parametr sterujący przenosi część wartości nagłówka HTTP Content-Type)	Jeśli jest obecny, jest zapisywany w komunikacie jako część nagłówka Content-Type. Uwaga: Wartość tego elementu sterującego powinna być udostępniana przez procedurę obsługi danych lub powiązanie danych.	Odczytywany z komunikatu odpowiedzi, nagłówka Content-Type	Odczytywany z komunikatu żądania, nagłówka Content-Type	Jeśli jest obecny, jest zapisywany w komunikacie jako część nagłówka Content-Type. Uwaga: Wartość tego elementu sterującego powinna być udostępniana przez procedurę obsługi danych lub powiązanie danych.
Character Set (zestaw znaków, domyślnie: UTF-8)	Jeśli jest obecny, jest zapisywany w komunikacie jako część nagłówka Content-Type. Uwaga: Wartość tego elementu sterującego powinna być udostępniana przez powiązanie danych.	Odczytywany z komunikatu odpowiedzi, nagłówka Content-Type	Odczytywany z komunikatu żądania, nagłówka Content-Type	Obsługiwany: zapisywany w komunikacie jako część nagłówka Content-Type. Uwaga: Wartość tego elementu sterującego powinna być udostępniana przez powiązanie danych.

Tabela 63. Dostarczane informacje nagłówka HTTP (kontynuacja)

Nazwa elementu sterującego	Żądanie importu HTTP	Odpowiedź importu HTTP	Żądanie eksportu HTTP	Odpowiedź eksportu HTTP
Transfer Encoding (kodowanie przesyłanych danych, możliwe wartości: chunked, identity; wartość domyślna: identity)	Jeśli jest obecny, jest zapisywany w komunikacie jako nagłówek i steruje kodowaniem transformowania komunikatu.	Odczytywany z komunikatu odpowiedzi	Odczytywany z komunikatu żądania	Jeśli jest obecny, jest zapisywany w komunikacie jako nagłówek i steruje kodowaniem transformowania komunikatu.
Content Encoding (kodowanie treści, możliwe wartości: gzip, x-gzip, deflate, identity; wartość domyślna: identity)	Jeśli jest obecny, jest zapisywane w komunikacie jako nagłówek i steruje kodowaniem ładunku.	Odczytywany z komunikatu odpowiedzi	Odczytywany z komunikatu żądania	Jeśli jest obecny, jest zapisywane w komunikacie jako nagłówek i steruje kodowaniem ładunku.
Content-Length (długość treści)	Zignorowano	Odczytywany z komunikatu odpowiedzi	Odczytywany z komunikatu żądania	Zignorowano
StatusCode (kod statusu, wartość domyślna: 200)	Nieobsługiwane	Odczytywany z komunikatu odpowiedzi	Nieobsługiwane	Jeśli jest obecny, jest zapisywany w komunikacie w wierszu odpowiedzi.
ReasonPhrase (frazą przyczyny, wartość domyślna: OK)	Nieobsługiwane	Odczytywany z komunikatu odpowiedzi	Nieobsługiwane	Wartość kontrolna jest ignorowana. Wartość wiersza odpowiedzi komunikatu jest generowana na podstawie kodu statusu.
Authentication (uwierzytelnianie, zawiera wiele właściwości)	Jeśli jest obecny, jest używany do konstruowania nagłówka podstawowego uwierzytelnienia. Uwaga: Wartość tego nagłówka będzie kodowana tylko w protokole HTTP. W architekturze SCA będzie on dekodowany i przekazywany w formie jawnego tekstu.	Nie dotyczy	Odczytywany z nagłówka podstawowego uwierzytelnienia komunikatu żądania. Obecność tego nagłówka nie oznacza, że użytkownik został uwierzytelniony. Uwierzytelnianie powinno być kontrolowane w konfiguracji serwletu. Uwaga: Wartość tego nagłówka będzie kodowana tylko w protokole HTTP. W architekturze SCA będzie on dekodowany i przekazywany w formie jawnego tekstu.	Nie dotyczy
Proxy (zawiera wiele właściwości: Host, Port, Authentication - uwierzytelnianie)	Jeśli jest obecny, jest używany do nawiązywania połączenia za pośrednictwem serwera proxy.	Nie dotyczy	Nie dotyczy	Nie dotyczy

Tabela 63. Dostarczane informacje nagłówka HTTP (kontynuacja)

Nazwa elementu sterującego	Żądanie importu HTTP	Odpowiedź importu HTTP	Żądanie eksportu HTTP	Odpowiedź eksportu HTTP
SSL (zawiera wiele właściwości: Keystore - magazyn kluczy, Keystore Password - hasło magazynu kluczy, Trustore - zaufany magazyn kluczy, Trustore Password - hasło zaufanego magazynu kluczy, ClientAuth - uwierzytelnianie klienta)	Jeśli jest wypełniony, a docelowy adres URL jest adresem protokołu HTTPS, jest używany do nawiązania połączenia za pośrednictwem protokołu SSL.	Nie dotyczy	Nie dotyczy	Nie dotyczy

Powiązania danych HTTP:

Dla każdego odwzorowania danych między komunikatem SCA (Service Component Architecture) i komunikatem protokołu HTTP należy skonfigurować procedurę obsługi danych lub powiązanie danych HTTP. Procedury obsługi danych udostępniają interfejs niezależny od powiązania, który może być ponownie wykorzystywany w powiązaniach transportu i stanowi rekomendowane podejście. Powiązania danych są specyficzne dla konkretnego powiązania transportu. Klasy powiązań danych specyficznych dla protokołu HTTP są dostarczane wraz z produktem. Użytkownik może również tworzyć niestandardowe procedury obsługi danych lub powiązania danych.

Uwaga: Trzy klasy powiązań danych HTTP opisywane w niniejszym temacie (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML i HTTPServiceGatewayDataBinding) są nieaktualne w produkcie IBM Business Process Manager 7.0. Zamiast używać powiązań danych opisanych w tym temacie warto rozważyć zastosowanie następujących procedur obsługi danych:

- SOAPDataHandler zamiast HTTPStreamDataBindingSOAP,
- UTF8XMLDataHandler zamiast HTTPStreamDataBindingXML,
- GatewayTextDataHandler zamiast HTTPServiceGatewayDataBinding.

Do używania z importami i eksportami HTTP udostępniane są powiązania danych: powiązanie danych binarnych, powiązanie danych XML i powiązanie danych SOAP. Powiązanie danych odpowiedzi nie jest wymagane w przypadku operacji jednokierunkowych. Powiązanie danych jest reprezentowane przez nazwę klasy Java, której instancje mogą dokonywać konwersji z protokołu HTTP na obiekt ServiceDataObject i z powrotem. W eksporcie należy użyć selektora funkcji, który w połączeniu z powiązaniem metod może określić używane powiązanie danych i wywołaną operację. Dostarczane są następujące powiązania danych:

- Powiązania danych binarnych, które traktują treść jako nieustrukturyzowane dane binarne. Implementacja schematu XSD powiązania danych binarnych jest następująca:

```
<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>
```

```
        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

- Powiązania danych XML, które obsługują treść jako dane XML. Implementacja powiązania danych XML jest podobna do powiązania danych XML JMS i nie ma żadnych ograniczeń schematu interfejsu.
- Powiązania danych SOAP, które obsługują treść jako dane SOAP. Implementacja powiązania danych SOAP nie ma ograniczeń schematu interfejsu.

Implementowanie niestandardowych powiązań danych HTTP

Ta sekcja opisuje, w jaki sposób implementowane są niestandardowe powiązania danych HTTP.

Uwaga: Zalecany podejściem jest implementowanie niestandardowej procedury obsługi danych, ponieważ może być ona ponownie wykorzystywana w powiązaniach transportu.

Podstawowym interfejsem do obsługi niestandardowych komunikatów HTTP jest interfejs `HTTPStreamDataBinding`. Interfejs ten jest zaprojektowany tak, aby umożliwić obsługę dużych ładunków. Jednak aby taka implementacja działała, powiązanie danych musi zwracać informacje kontrolne i nagłówki przed zapisywaniem komunikatu do strumienia.

Niestandardowe powiązanie danych musi implementować metody i kolejność wykonywania zgodnie z poniższą listą.

Aby dostosować powiązanie danych, należy napisać klasę implementującą interfejs `HTTPStreamDataBinding`. Powiązanie danych powinno mieć cztery właściwości prywatne:

- `private DataObject pDataObject`,
- `private HTTPControl pCtrl`,
- `private HTTPHeaders pHeaders`,
- `private yourNativeDataType nativeData`.

Powiązanie HTTP będzie wywoływało niestandardowe powiązanie danych w następującej kolejności:

- Przetwarzanie danych wychodzących (obiekt `DataObject` do formatu rodzimego):
 1. `setDataObject(...)`
 2. `setHeaders(...)`
 3. `setControlParameters(...)`
 4. `setBusinessException(...)`
 5. `convertToNativeData()`
 6. `getControlParameters()`
 7. `getHeaders()`
 8. `write(...)`
- Przetwarzanie danych przychodzących (format rodzimy do obiektu `DataObject`):
 1. `setControlParameters(...)`
 2. `setHeaders(...)`
 3. `convertFromNativeData(...)`
 4. `isBusinessException()`
 5. `getDataObject()`
 6. `getControlParameters()`
 7. `getHeaders()`

Metoda `setDataObject(...)` musi być wywołana w metodzie `convertFromNativeData(...)`, aby ustawić wartość obiektu `dataObject`, który jest przekształcany z danych rodzimych we właściwość prywatną `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;
}
/*
 * Dodawanie nagłówka HTTP IsBusinessException w nagłówkach pHeaders.
 * Dwa kroki:
 * 1.Usunięcie wszystkich nagłówków o nazwie IsBusinessException (bez rozróżniania wielkości liter).
 *   Ma to na celu upewnienie się, że istnieje tylko jeden taki nagłówek.
 * 2.Dodanie nowego nagłówka IsBusinessException.
 */
public void setBusinessException(boolean isBusinessException) {
    //Usunięcie wszystkich nagłówków o nazwie IsBusinessException (bez rozróżniania wielkości liter).
    //Ma to na celu upewnienie się, że istnieje tylko jeden taki nagłówek.
    //Dodanie nowego nagłówka IsBusinessException. Przykład kodu:
    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;
}
/*
 * Uzyskanie nagłówka IsBusinessException z nagłówków pHeaders, zwrócenie jego wartości boolowskiej.
 */
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){
    //Metoda opracowana przez klienta:
    //Odczytywanie danych z interfejsu HTTPInputStream.
    //Przekształcenie danych w obiekt DataObject.
    DataObject dataobject=realConvertWorkFromNativeDataToSD0(arg0);
    setDataObject(dataobject);
}

public void write(HTTPOutputStream output) throws IOException {
    if (nativeData != null)
        output.write(nativeData);
}

```

Powiązania EJB

Powiązania importu EJB (Enterprise JavaBeans) umożliwiają komponentom SCA (Service Component Architecture) wywoływanie usług udostępnianych przez logikę biznesową Java EE działającą na serwerze Java EE. Powiązania

eksportu EJB umożliwiają ujawnianie komponentów SCA jako komponentów EJB (Enterprise JavaBeans), co pozwala logice biznesowej Java EE wywoływać komponenty SCA, które są w przeciwnym razie dla nich niedostępne.

Powiązania importu EJB:

Powiązania importu EJB umożliwiają modułowi SCA wywoływanie implementacji EJB poprzez określenie sposobu, w jaki moduł konsumenta jest powiązany z zewnętrznym komponentem EJB. Importowanie usług z zewnętrznej implementacji EJB umożliwia użytkownikom podłączenie logiki biznesowej do środowiska produktu IBM Business Process Manager i uczestnictwo w procesie biznesowym.

Do tworzenia powiązań importu EJB służy produkt Integration Designer. Do wygenerowania powiązań można użyć jednej z poniższych procedur:

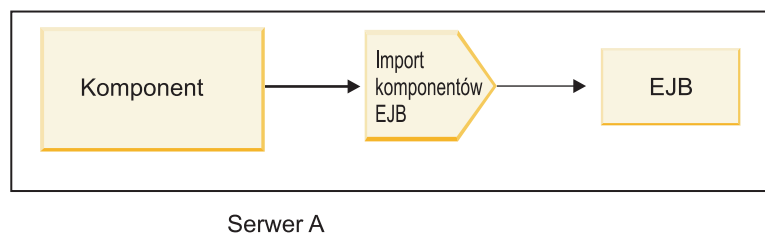
- Tworzenie importu EJB przy użyciu kreatora usług zewnętrznych
Do zbudowania importu EJB na podstawie istniejącej implementacji można użyć kreatora usług zewnętrznych w produkcie Integration Designer. Kreator usług zewnętrznych umożliwia tworzenie usług na podstawie podawanych przez użytkownika kryteriów. Generuje on następnie obiekty biznesowe, interfejsy oraz pliki importu w oparciu o wykryte usługi.
- Tworzenie importu EJB przy użyciu edytora składania
Istnieje możliwość utworzenia importu EJB w diagramie składania przy użyciu edytora składania produktu Integration Designer. Do utworzenia powiązania EJB przy użyciu palety można użyć importu lub klasy Java.

Wygenerowany import ma powiązania danych, które nawiązują połączenie Java-WSDL i nie wymagają komponentu pomocowego języka Java. Komponent można połączyć bezpośrednio z odwołaniem WSDL (Web Services Description Language) prowadzącym do importu EJB komunikującego się z usługą opartą na komponencie EJB przy użyciu interfejsu Java.

Import EJB może realizować interakcję z logiką biznesową Java EE przy użyciu modelu programowania EJB 2.1 lub modelu programowania EJB 3.0.

Wywołanie logiki biznesowej Java EE może być lokalne (tylko dla modelu EJB 3.0) lub zdalne.

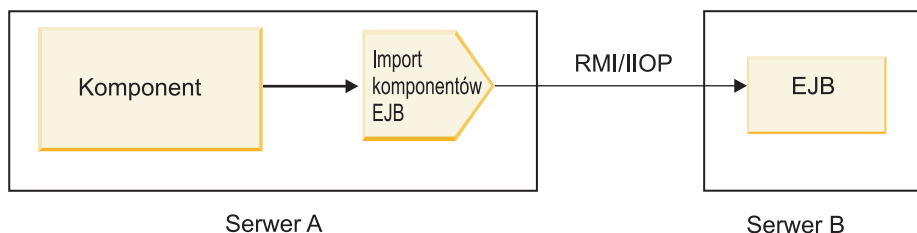
- Wywołanie lokalne jest używane, jeśli istnieje potrzeba wywołania logiki biznesowej Java EE znajdującej się na tym samym serwerze co import.



Rysunek 73. Wywołanie lokalne komponentu EJB (tylko dla modelu EJB 3.0)

- Wywołanie zdalne jest używane, jeśli istnieje potrzeba wywołania logiki biznesowej Java EE, która nie znajduje się na tym samym serwerze co import.

Na poniższej przykładowej ilustracji import EJB używa zdalnego wywołania metody za pośrednictwem protokołu Internet InterORB (RMI/IIOP) w celu wywołania metody EJB na innym serwerze.



Rysunek 74. Zdalne wywołanie komponentu EJB

Podczas konfigurowania powiązania EJB produkt Integration Designer używa nazwy JNDI, aby określić poziom modelu programowania komponentu EJB oraz typ wywołania (lokalne lub zdalne).

Powiązania importu EJB zawierają następujące główne komponenty:

- procedurę obsługi danych JAX-WS,
- selektor błędu komponentu EJB,
- selektor funkcji importu EJB.

Jeśli scenariusz użytkownika nie jest oparty na odwzorowaniu JAX-WS, do wykonania czynności realizowanych w innych przypadkach przez komponenty stanowiące część powiązań importu EJB mogą być potrzebne niestandardowa procedura obsługi danych, selektor funkcji i selektor błędu. Dotyczy to również odwzorowania zazwyczaj realizowanego przez niestandardowy algorytm odwzorowania.

Powiązania eksportu EJB:

Zewnętrzne aplikacje Java EE mogą wywoływać komponent SCA za pośrednictwem powiązania eksportu EJB. Jeśli wykorzystywany jest eksport EJB, istnieje możliwość ujawniania komponentów SCA tak, aby zewnętrzne aplikacje Java EE mogły wywoływać te komponenty przy użyciu modelu programowania EJB.

Uwaga: eksport komponentu EJB jest bezstanowym komponentem bean.

Do tworzenia powiązań EJB służy produkt Integration Designer. Do wygenerowania powiązań można użyć jednej z poniższych procedur:

- Tworzenie powiązań eksportu EJB przy użyciu kreatora usług zewnętrznych
Do zbudowania usługi eksportu EJB na podstawie istniejącej implementacji można użyć kreatora usług zewnętrznych w produkcie Integration Designer. Kreator usług zewnętrznych umożliwia tworzenie usług na podstawie podawanych przez użytkownika kryteriów. Generuje on następnie obiekty biznesowe, interfejsy oraz pliki eksportu w oparciu o wykryte usługi.
- Tworzenie powiązań eksportu EJB przy użyciu edytora składania
Istnieje możliwość utworzenia eksportu EJB przy użyciu edytora składania produktu Integration Designer.

Ważne: klient J2SE (Java 2 Platform, Standard Edition) nie może wywoływać klienta eksportu EJB wygenerowanego w produkcie Integration Designer.

Użytkownik może wygenerować powiązanie z istniejącego komponentu SCA lub wygenerować eksport z powiązaniem EJB dla interfejsu Java.

- Jeśli użytkownik generuje eksport dla istniejącego komponentu SCA zawierającego istniejący interfejs WSDL, do eksportu przypisywany jest interfejs Java.
- Jeśli użytkownik generuje eksport dla interfejsu Java, można wybrać interfejs WSDL lub interfejs Java dla eksportu.

Uwaga: interfejs Java używany do tworzenia eksportu EJB charakteryzuje się następującymi ograniczeniami obiektów (parametrów wejściowych i wyjściowych oraz wyjątków), które są przekazywane jako parametry w wywołaniu zdalnym:

- Obiekty te muszą mieć konkretny typ (zamiast typu interfejsu lub typu abstrakcyjnego).
- Obiekty te muszą być zgodne ze specyfikacją EJB (Enterprise JavaBeans). Musi być możliwe ich serializowanie i muszą mieć domyślny konstruktor no-argument, zaś wszystkie ich właściwości muszą być dostępne za pośrednictwem metod procedur pobierających i procedur ustawiających.

Informacje na temat specyfikacji Enterprise JavaBeans zawiera serwis WWW firmy Sun Microsystems Inc. - <http://java.sun.com>.

Dodatkowo wyjątek musi być wyjątkiem sprawdzonym, dziedziczonym z klasy `java.lang.Exception` i musi być pojedynczy (czyli nie może obsługiwać zgłaszania wielu sprawdzonych typów wyjątków).

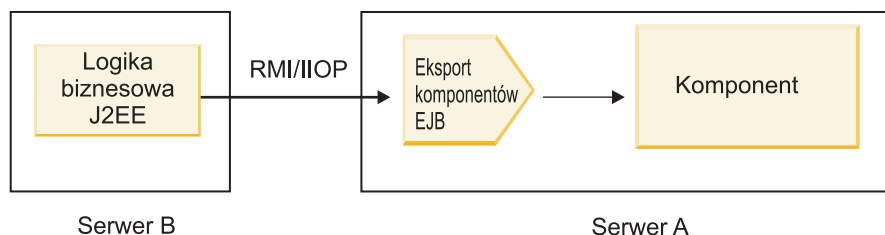
Należy również zauważyć, że interfejs biznesowy komponentu EJB języka Java jest zwykłym interfejsem Java i nie może rozszerzać interfejsu `javax.ejb.EJBObject` ani interfejsu `javax.ejb.EJBLocalObject`. Metody interfejsu biznesowego nie powinny zgłaszać wyjątku `java.rmi.Remote.Exception`.

Powiązania eksportu EJB mogą realizować interakcję z logiką biznesową Java EE przy użyciu modelu programowania EJB 2.1 lub modelu programowania EJB 3.0.

Wywołanie może być lokalne (tylko dla modelu EJB 3.0) lub zdalne.

- Wywołanie lokalne jest używane, jeśli logika biznesowa Java EE wywołuje komponent SCA znajdujący się na tym samym serwerze co eksport.
- Wywołanie zdalne jest używane, jeśli logika biznesowa Java EE nie znajduje się na tym samym serwerze co eksport.

Na przykład na poniższej ilustracji komponent EJB używa technologii RMI/IIOP do wywoływania komponentu SCA na innym serwerze.



Rysunek 75. Wywołanie zdalne z klienta do komponentu SCA za pośrednictwem eksportu EJB

Podczas konfigurowania powiązania EJB produkt Integration Designer używa nazwy JNDI, aby określić poziom modelu programowania komponentu EJB oraz typ wywołania (lokalne lub zdalne).

Powiązania eksportu EJB zawierają następujące główne komponenty:

- procedurę obsługi danych JAX-WS,
- selektor funkcji eksportu EJB.

Jeśli scenariusz użytkownika nie jest oparty na odwzorowaniu JAX-WS, do wykonania czynności realizowanych w innych przypadkach przez komponenty stanowiące część powiązań eksportu EJB mogą być potrzebne niestandardowa procedura obsługi danych i selektor funkcji. Dotyczy to również odwzorowania zazwyczaj realizowanego przez niestandardowy algorytm odwzorowania.

Właściwości powiązania EJB:

Powiązania importu EJB używają skonfigurowanych nazw JNDI do określenia poziomu modelu programowania komponentu EJB oraz typu wywołania (lokalne lub zdalne). Powiązania importu i eksportu EJB używają procedury obsługi danych JAX-WS do transformowania danych. Powiązanie importu EJB używa selektora funkcji importu EJB i selektora błędu EJB, a powiązanie eksportu EJB używa selektora funkcji eksportu EJB.

Nazwy JNDI i powiązania importu EJB:

Podczas konfigurowania powiązania EJB dla importu produkt Integration Designer używa nazwy JNDI, aby określić poziom modelu programowania komponentu EJB oraz typ wywołania (lokalne lub zdalne).

Jeśli nie zostanie określona nazwa JNDI, używane jest domyślne powiązanie interfejsu EJB. Tworzone nazwy domyślne zależą od tego, czy wywoływany jest komponent JavaBeans standardu EJB 2.1, czy komponent JavaBeans standardu EJB 3.0.

Uwaga: Więcej informacji na temat konwencji nazewnictwa zawiera temat „Przegląd powiązań aplikacji EJB 3.0” dostępny w Centrum informacyjnym produktu WebSphere Application Server.

- Komponenty JavaBean standardu EJB 2.1

Domyślna nazwa JNDI wstępnie wybrana przez produkt Integration Designer to domyślne powiązanie EJB 2.1, które przyjmuje postać łańcucha **ejb/** skonkatenowanego z interfejsem głównym rozdzielanym ukośnikami.

Na przykład w przypadku interfejsu głównego komponentu JavaBeans w standardzie EJB 2.1 `com.mojafirma.mojzdalnybiznes` domyślnym powiązaniem jest:

```
ejb/com/mojafirma/mojzdalnybiznes
```

W przypadku standardu EJB 2.1 obsługiwane jest tylko zdalne wywoływanie komponentów EJB.

- Komponenty JavaBeans standardu EJB 3.0

Domyślna nazwa JNDI wstępnie wybrana przez produkt Integration Designer dla lokalnej nazwy JNDI jest pełną nazwą klasy interfejsu lokalnego poprzedzoną łańcuchem **ejblocal:**. Na przykład wstępnie wybraną nazwą JNDI w standardzie EJB 3.0 dla pełnego interfejsu lokalnego `com.mojafirma.mojbizneslokalny` jest:

```
ejblocal:com.mojafirma.mojbizneslokalny
```

W przypadku zdalnego interfejsu `com.mojafirma.mojbizneszdalny` wybraną nazwą JNDI w standardzie EJB 3.0 jest pełny interfejs:

```
com.mojafirma.mojbizneszdalny
```

Domyślne powiązania aplikacji EJB 3.0 są opisane w następującym miejscu: EJB 3.0 application bindings overview (Przegląd powiązań aplikacji EJB 3.0).

Jako domyślnego położenia JNDI dla komponentów EJB korzystających z modelu programowania w wersji 3.0 produkt Integration Designer używa nazwy skróconej.

Uwaga: Jeśli z powodu zastosowania lub skonfigurowania niestandardowego odwzorowania wdrożone odwołanie JNDI dla docelowego komponentu EJB jest różne niż domyślne położenie powiązania JNDI, należy poprawnie określić docelową nazwę JNDI. Nazwę tę można określić w produkcie Integration Designer przed wdrożeniem, a w przypadku powiązania importu można ją również zmienić w Konsoli administracyjnej (po wdrożeniu) tak, aby była zgodna z nazwą JNDI docelowego komponentu EJB.

Więcej informacji na temat tworzenia powiązań EJB zawiera sekcja poświęcona pracy z powiązaniem EJB w Centrum informacyjnym produktu Integration Designer.

Procedura obsługi danych JAX-WS:

Powiązanie importu EJB (Enterprise JavaBeans) stosuje procedurę obsługi danych JAX-WS do przekształcania obiektów biznesowych żądania w parametry obiektu Java oraz do przekształcania wartości zwracanej obiektu Java w obiekt biznesowy odpowiedzi. Powiązanie eksportu EJB używa procedury obsługi danych JAX-WS do przekształcania komponentów EJB żądania w obiekty biznesowe żądania oraz do przekształcania obiektu biznesowego odpowiedzi w wartość zwracaną.

Ta procedura obsługi danych odwzorowuje dane z interfejsu WSDL określonego w architekturze SCA na docelowy interfejs Java komponentu EJB (i odwrotnie). Wykorzystuje w tym celu specyfikację JAX-WS (Java API for XML Web Services) i specyfikację JAXB (Java Architecture for XML Binding).

Uwaga: Aktualnie obsługa jest ograniczona do specyfikacji JAX-WS 2.1.1 i JAXB 2.1.3.

Procedura obsługi danych określona na poziomie powiązania EJB jest używana do przetwarzania żądań, odpowiedzi, błędów i wyjątków czasu wykonywania.

Uwaga: W przypadku błędów istnieje możliwość określenia specyficznej procedury obsługi danych dla każdego błędu poprzez określenie właściwości konfiguracji `faultBindingType`. Powoduje to nadpisanie wartości określonej na poziomie powiązania EJB.

Procedura obsługi danych JAX-WS jest używana domyślnie, jeśli powiązanie EJB ma interfejs WSDL. Tej procedury obsługi danych nie można używać do transformowania komunikatu SOAP reprezentującego wywołanie JAX-WS na obiekt danych.

Powiązanie importu EJB używa procedury obsługi danych do transformowania obiektu danych w tablicę `Object` języka Java (`Object[]`). Podczas komunikacji wychodzącej realizowane są następujące działania przetwarzania:

1. Powiązanie EJB ustawia oczekiwany typ, oczekiwany element i nazwę metody docelowej we właściwości `BindingContext` tak, aby były zgodne z danymi określonymi w pliku WSDL.
2. Powiązanie EJB wywołuje metodę transformowania dla obiektu danych, który wymaga transformowania danych.
3. Procedura obsługi danych zwraca tablicę `Object[]` reprezentującą parametry metody (w kolejności ich definiowania wewnątrz metody).
4. Powiązanie EJB używa tablicy `Object[]` do wywołania metody dla docelowego interfejsu EJB.

Powiązanie przygotowuje również tablicę `Object[]` tak, aby przetwarzać odpowiedź z wywołania komponentu EJB.

- Pierwszy element w tablicy `Object[]` jest wartością zwracaną z wywołania metody Java.
- Kolejne wartości reprezentują parametry wejściowe dla metody.

Jest to wymagane do obsługi typów parametrów `In/Out` i `Out`.

W przypadku parametrów typu `Out` wartości muszą być zwracane w obiekcie danych odpowiedzi.

Procedura obsługi danych przetwarza i transformuje wartości znalezione w tablicy `Object[]`, a następnie zwraca odpowiedź do obiektu danych.

Procedura obsługi danych obsługuje elementy `xs:AnyType`, `xs:AnySimpleType` i `xs:Any`, a także inne typy danych XSD. Aby włączyć obsługę elementu `xs:Any`, należy użyć wartości `@XmlElement (lax=true)` dla właściwości komponentu `JavaBeans` w kodzie Java. Zostało to przedstawione w poniższym przykładzie:

```
public class TestType {
    private Object[] object;

    @XmlElement (lax=true)
    public Object[] getObject() {
        return object;
    }

    public void setObject (Object[] object) {
        this.object=object;
    }
}
```

Ten kod powoduje, że obiekt właściwości w klasie `TestType` jest polem `xs:any`. Wartość klasy Java używanej w polu `xs:any` powinna mieć adnotację `@XmlElement`. Jeśli na przykład `Address` jest klasą Java używaną do zapełniania tablicy obiektów, klasa `Address` powinna mieć adnotację `@XmlElement`.

Uwaga: Aby dostosować odwzorowanie z typu XSD na typy Java zdefiniowane przez specyfikację JAX-WS, należy zmienić adnotacje JAXB tak, aby spełniały potrzeby biznesowe. Procedura obsługi danych JAX-WS obsługuje elementy `xs:any`, `xs:anyType`, `xs:anySimpleType`.

Procedura obsługi danych JAX-WS podlega następującym ograniczeniom:

- Procedura obsługi danych nie obejmuje obsługi adnotacji **@WebParam** atrybutu nagłówka.
- Przestrzeń nazw dla plików schematu obiektu biznesowego (plików XSD) nie obejmuje domyślnego odwzorowania z nazwy pakietu Java. Adnotacja **@XMLSchema** w klasie `package-info.java` również nie działa. Jedynym sposobem utworzenia definicji XSD z przestrzenią nazw jest użycie adnotacji **@XmlType** i **@XmlRootElement**. Adnotacja **@XmlRootElement** definiuje docelową przestrzeń nazw dla elementu globalnego w typach komponentów JavaBeans.
- Kreator importu EJB nie tworzy plików XSD dla klas niepowiązanych. Wersja 2.0 nie obsługuje adnotacji **@XmlSeeAlso**, a więc jeśli klasa potomna nie jest przywoływana bezpośrednio z klasy macierzystej, nie zostanie utworzona definicja XSD. Rozwiązaniem tego problemu jest uruchomienie programu narzędziowego SchemaGen dla takich klas potomnych.

Program SchemaGen jest programem narzędziowym wiersza komend (znajdującym się w katalogu `katalog_główny_instalacji_produktu_WPS/bin`), udostępnianym w celu tworzenia plików XSD dla określonego komponentu bean. Te pliki XSD należy ręcznie skopiować do modułu, aby rozwiązanie zadziałało.

Selektor błędów komponentu EJB:

Selektor błędów komponentu EJB określa, czy wywołanie komponentu EJB zakończyło się błędem, wyjątkiem czasu wykonywania czy pomyślną odpowiedzią.

Jeśli został wykryty błąd, selektor błędów komponentu EJB zwraca rodzimą nazwę błędu do środowiska wykonawczego powiązania, aby procedura obsługi danych JAX-WS mogła przekształcić obiekt wyjątku w obiekt biznesowy błąd.

W przypadku pomyślnej odpowiedzi (niebędącej błędem), powiązanie importu EJB składa tablicę obiektów Java (`Object[]`) w celu zwrócenia wartości.

- Pierwszy element w tablicy `Object[]` jest wartością zwracaną z wywołania metody Java.
- Kolejne wartości reprezentują parametry wejściowe dla metody.

Jest to wymagane do obsługi typów parametrów In/Out i Out.

W przypadku scenariuszy wyjątków, powiązanie składa tablicę `Object[]`, a jej pierwszy element reprezentuje wyjątek zgłoszony przez metodę.

Selektor błędów może zwrócić jedną spośród następujących wartości:

Tabela 64. Wartości zwracane

Typ	Wartość zwracana	Opis
Błąd	ResponseType.FAULT	Wartość zwracana, jeśli przekazana tablica <code>Object[]</code> zawiera obiekt wyjątku.
Wyjątek czasu wykonywania	ResponseType.RUNTIME	Wartość zwracana, jeśli obiekt wyjątku nie jest zgodny z żadnym spośród zadeklarowanych typów wyjątku w metodzie.
Normalna odpowiedź	ResponseType.RESPONSE	Wartość zwracana we wszystkich pozostałych przypadkach.

Jeśli selektor błędów zwróci wartość **ResponseType.FAULT**, zwracana jest rodzima nazwa błędu. Ta rodzima nazwa błędu jest używana przez powiązanie w celu określenia odpowiedniej nazwy błędu WSDL z modelu oraz wywołania poprawnej procedury obsługi danych błędu.

Selektor funkcji komponentu EJB:

Powiązania EJB używają selektora funkcji importu (przy przetwarzaniu danych wychodzących) lub selektora funkcji eksportu (przy przetwarzaniu danych przychodzących), aby określić, która metoda EJB ma zostać wywołana.

Selektor funkcji importu

W przypadku przetwarzania danych wychodzących selektor funkcji importu określa typ metody EJB na podstawie nazwy operacji wywołanej przez komponent SCA połączony z importem EJB. Selektor funkcji wyszukuje adnotację `@WebMethod` w wygenerowanej przez produkt Integration Designer klasie Java z adnotacją JAX-WS, aby określić nazwę powiązanej operacji docelowej.

- Jeśli występuje adnotacja `@WebMethod`, selektor funkcji używa adnotacji `@WebMethod` do określenia poprawnego odwzorowania metody Java dla metody WSDL.
- Jeśli nie występuje adnotacja `@WebMethod`, selektor funkcji zakłada, że nazwa metody Java jest taka sama jak nazwa wywołanej operacji.

Uwaga: Ten selektor funkcji działa tylko w przypadku interfejsu typu WSDL w imporcie EJB, ale nie działa w przypadku interfejsu typu Java w imporcie EJB.

Selektor funkcji zwraca obiekt `java.lang.reflect.Method` reprezentujący metodę interfejsu EJB.

Selektor funkcji używa tablicy `Object` języka Java (`Object[]`), w której zawierana jest odpowiedź z metody docelowej. Pierwszy element tablicy `Object[]` jest metodą Java o nazwie pliku WSDL, a drugi element w tablicy `Object[]` jest wejściowym obiektem biznesowym.

Selektor funkcji eksportu

W przypadku przetwarzania danych wychodzących selektor funkcji eksportu określa, która metoda docelowa ma zostać wykonana, na podstawie metody Java.

Selektor funkcji eksportu odwzorowuje nazwę operacji Java wywołanej przez klienta EJB na nazwę operacji w interfejsie komponentu docelowego. Nazwa metody jest zwracana w postaci łańcucha i rozstrzygana przez środowisko wykonawcze SCA w zależności od typu interfejsu komponentu docelowego.

Powiązania EIS

Powiązania EIS (Enterprise Information System) udostępniają połączenia między komponentami SCA i zewnętrznym systemem EIS. Ta komunikacja jest realizowana za pomocą eksportów EIS i importów EIS, które obsługują adaptory zasobów architektury JCA 1.5 i adaptory WebSphere Adapters.

Komponenty SCA mogą wymagać przesyłania danych do zewnętrznego systemu EIS lub z takiego systemu. Podczas tworzenia modułu SCA wymagającego takich połączeń należy dołączyć (oprócz komponentu SCA) import lub eksport z powiązaniem EIS umożliwiającym komunikowanie z określonym zewnętrznym systemem EIS.

Adaptory zasobów w produkcie IBM Integration Designer są używane w kontekście importu lub eksportu. Importy i eksporty są opracowywane za pomocą kreatora usług zewnętrznych, a w trakcie ich opracowywania dołączany jest adapter zasobu. Import EIS, który umożliwia aplikacji wywoływanie usługi w systemie EIS, lub eksport EIS, który umożliwia aplikacji w systemie EIS wywoływanie usługi utworzonej w produkcie IBM Integration Designer, są tworzone przy użyciu adaptera zasobu. Na przykład w celu wywołania usługi w systemie JD Edwards należy utworzyć import z adapterem systemu JD Edwards.

W przypadku używania kreatora usług zewnętrznych informacje powiązania EIS są tworzone dla użytkownika. Do dodawania lub modyfikowania informacji o powiązaniu można również użyć innego narzędzia - edytora składania. Więcej informacji zawiera sekcja Uzyskiwanie dostępu do usług zewnętrznych za pomocą adapterów.

Po wdrożeniu modułu SCA zawierającego powiązanie EIS można użyć Konsoli administracyjnej, aby wyświetlić informacje o powiązaniu lub aby je skonfigurować.

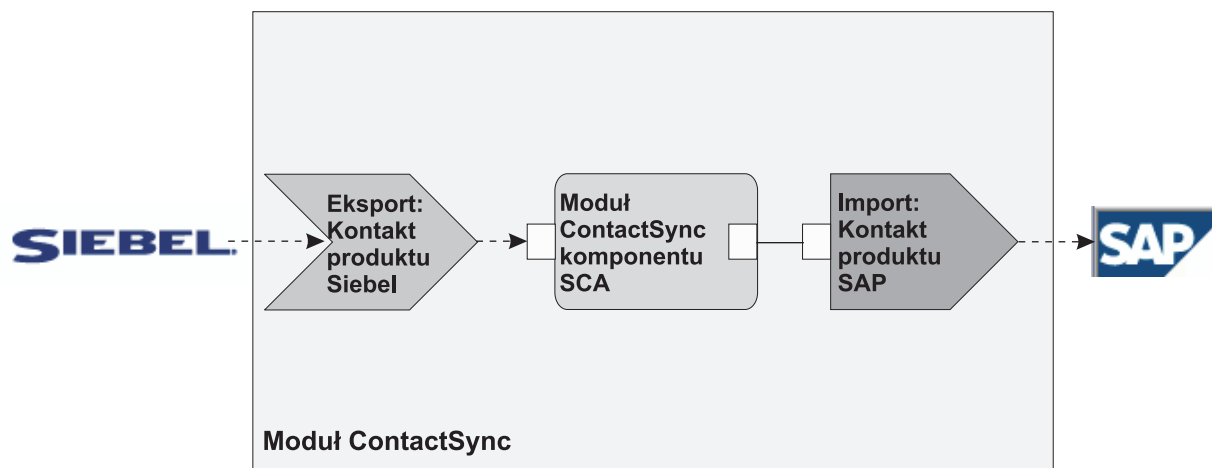
Przegląd powiązań EIS:

Powiązanie EIS (Enterprise Information System) używane z adapterem zasobu JCA umożliwia dostęp do usług w systemie EIS lub udostępnienie usług systemowi EIS.

Poniższy przykład przedstawia sposób, w jaki moduł SCA o nazwie ContactSyncModule synchronizuje informacje kontaktowe między systemem Siebel i systemem SAP.

1. Komponent SCA o nazwie ContactSync nasłuchuje (za pośrednictwem eksportu aplikacji EIS o nazwie Siebel Contact) zmian w kontaktach systemu Siebel.
2. Komponent SCA ContactSync korzysta z aplikacji SAP (za pośrednictwem importu aplikacji EIS), aby zaktualizować w odpowiedni sposób informacje kontaktowe systemu SAP.

Ponieważ struktury danych używane do przechowywania kontaktów są różne w systemach Siebel i SAP, komponent SCA ContactSync musi udostępniać odwzorowanie.



Rysunek 76. Przepływ z systemu Siebel do systemu SAP

Eksport kontaktów systemu Siebel i import kontaktów systemu SAP mają skonfigurowane odpowiednie adaptory zasobów.

Kluczowe opcje powiązań z systemami EIS:

Element importowany EIS jest elementem importowanym architektury Service Component Architecture (SCA) umożliwiającym komponentom zawartym w module SCA na korzystanie z aplikacji EIS zdefiniowanych poza modulem SCA. Element importowany EIS jest używany do przesyłania danych z komponentu SCA do zewnętrznego systemu EIS, natomiast element eksportowany EIS jest używany do przesyłania danych z zewnętrznego systemu EIS do modułu SCA.

Elementy importowane

Element importowany EIS służy do łączenia komponentów architektury SCA z zewnętrznymi systemami EIS. Aplikacje zewnętrzne można traktować jak element importowany EIS. W tym przypadku element importowany EIS wysyła dane do zewnętrznego systemu EIS i opcjonalnie odbiera dane wysyłane do niego w odpowiedzi.

Element importowany EIS udostępnia komponentom SCA jednolity widok aplikacji zewnętrznych w stosunku do modułu. Pozwala to komponentom komunikować się z zewnętrznym systemem EIS, takim jak SAP, Siebel lub PeopleSoft, z użyciem spójnego modelu SCA.

Po stronie klienta elementu importowanego istnieje interfejs ujawniony przez aplikację importu EIS zawierający jedną lub kilka metod, z których każda pobiera obiekty danych jako argumenty i zwraca wartości. Po stronie implementacji występuje interfejs CCI (Common Client Interface) zaimplementowany przez adapter zasobów.

Implementacja wykonawcza importu EIS łączy interfejs po stronie klienta z interfejsem CCI. Import odwzorowuje wywołanie metody interfejsu w wywołanie po stronie CCI.

Powiązania są tworzone na trzech poziomach: powiązanie interfejsów korzystające z zawartych w nich powiązań metod, w których używane są powiązania danych.

Powiązanie interfejsów łączy interfejs elementu importowanego z połączeniem z systemem EIS udostępniającym aplikację. Odzwierciedla to fakt, że zestaw aplikacji reprezentowany przez interfejs jest udostępniany przez określoną instancję systemu EIS, a połączenie zapewnia dostęp do tej instancji. Element wiążący obejmuje właściwości udostępniające informacje, które są wystarczające do utworzenia połączenia (te właściwości są częścią instancji interfejsu `javax.resource.spi.ManagedConnectionFactory`).

Powiązanie metod łączy metodę z określoną interakcją z systemem EIS. W przypadku architektury JCA interakcja jest charakteryzowana przez zestaw właściwości implementacji interfejsu `javax.resource.cci.InteractionSpec`. Element interakcji powiązania metod obejmuje właściwości oraz nazwę klasy. Informacje te są wystarczające do wykonania interakcji. Powiązanie metod korzysta z powiązań danych opisujących odwzorowanie argumentu i wyniku metody interfejsu w reprezentację w systemie EIS.

Scenariusz wykonawczy związany z elementem importowanym EIS jest następujący:

1. Metoda interfejsu importu jest wywoływana z użyciem modelu programowania architektury SCA.
2. Żądanie docierające do importu EIS zawiera nazwę metody i jej argumenty.
3. Import tworzy najpierw implementację powiązania interfejsów, a następnie, korzystając z danych pochodzących z powiązania elementu importowanego, tworzy interfejs `ConnectionFactory` i wiąże te dwa elementy. Innymi słowy import wywołuje interfejs `setConnectionFactory` w powiązaniu interfejsów.
4. Utworzona zostaje implementacja powiązania metod odpowiadająca wywołanej metodzie.
5. Instancja interfejsu `javax.resource.cci.InteractionSpec` zostaje utworzona i zapełniona, a następnie z użyciem powiązań danych argumenty metody zostają przekształcone w format zrozumiały dla adaptera zasobów.
6. Za pomocą interfejsu CCI przeprowadzona zostaje interakcja.
7. Podczas powrotu z wywołania z użyciem powiązania danych utworzony zostaje wynik wywołania, a następnie wynik ten jest zwracany do programu wywołującego.

Eksperty

Ekspert EIS służy do łączenia komponentu architektury SCA z zewnętrznym systemem EIS. Aplikacje zewnętrzne można traktować jak element eksportowany EIS. W tym przypadku zewnętrzna aplikacja wysyła swoje dane w postaci okresowych powiadomień. Element eksportowany EIS można uważać za aplikację subskrypcji nasłuchującą zewnętrznych żądań z systemu EIS. Komponent SCA korzystający z elementu eksportowanego EIS widzi go jako aplikację lokalną.

Element eksportowany EIS udostępnia komponentom SCA jednolity widok aplikacji zewnętrznych w stosunku do modułu. Pozwala to komponentom komunikować się z systemem EIS, takim jak SAP, Siebel lub PeopleSoft, z użyciem spójnego modelu SCA.

Element eksportowany obejmuje implementację procesu nasłuchiwanie odbierającego żądania z systemu EIS. Proces nasłuchiwanie implementuje interfejs procesu nasłuchiwanie specyficznego dla danego adaptera zasobów. Element eksportowany obejmuje także komponent implementujący interfejs ujawniony systemowi EIS za pośrednictwem tego elementu eksportowanego.

Implementacja wykonawcza elementu eksportowanego EIS łączy proces nasłuchiwanie z komponentem implementującym interfejs. Element eksportowany odwzorowuje żądanie systemu EIS w wywołanie odpowiedniej operacji w komponentcie. Powiązania są tworzone na trzech poziomach: powiązanie procesu nasłuchiwanie korzystające z zawartego w nim powiązania metod własnych, w którym używane jest powiązanie danych.

Powiązanie procesu nasłuchiwanie łączy proces nasłuchiwanie odbierający żądania z komponentem ujawnionym za pośrednictwem elementu eksportowanego. Definicja elementu eksportowanego obejmuje nazwę komponentu. Środowisko wykonawcze znajduje ten komponent, a następnie przekazuje do niego żądania.

Powiązanie metod własnych wiąże metodę własną lub typ zdarzenia odebranego przez proces nasłuchiwanie z operacją implementowaną przez komponent ujawniony za pomocą elementu eksportowanego. Nie ma żadnego związku między metodą wywołaną w procesie nasłuchiwanie i typem zdarzenia. Wszystkie zdarzenia nadchodzą za pośrednictwem jednej lub kilku metod procesu nasłuchiwanie. Powiązanie metod własnych korzysta z selektora funkcji zdefiniowanego w elemencie eksportowanym w celu wyodrębnienia nazwy metody własnej z przychodzących danych, a powiązań danych w celu powiązania formatu danych systemu EIS z formatem zrozumiałym dla komponentu.

Scenariusz wykonawczy związany z elementem eksportowanym EIS jest następujący:

1. Żądanie systemu EIS powoduje wywołanie metody w implementacji procesu nasłuchiwanie.
2. Proces nasłuchiwanie odnajduje i wywołuje element eksportowany i przekazuje do niego wszystkie argumenty wywołania.
3. Element eksportowany tworzy implementację powiązania procesu nasłuchiwanie.
4. Element eksportowany tworzy instancję selektora funkcji i ustawia go na powiązanie procesu nasłuchiwanie.
5. Element eksportowany inicjuje powiązania metod własnych i dodaje je do powiązania procesu nasłuchiwanie. Dla każdego powiązania metod własnych zostają także zainicjowane powiązania danych.
6. Element eksportowany wywołuje powiązanie procesu nasłuchiwanie.
7. Powiązanie procesu nasłuchiwanie odnajduje komponenty i używa selektora funkcji do pobrania nazwy metody własnej.
8. Ta nazwa umożliwia odnalezienie powiązania metod własnych, które następnie wywołuje komponent docelowy.

Styl interakcji adaptera pozwala powiązaniu elementu eksportowanego EIS wywołać komponent docelowy asynchronicznie (opcja domyślna) lub synchronicznie.

Adaptory zasobów

Elementy importowane i eksportowane są opracowywane za pomocą kreatora usługi zewnętrznej, a w trakcie ich opracowywania dołączany jest adapter zasobów. Adaptory dostarczane z produktem IBM Integration Designer zapewniające dostęp do systemów CICS, IMS, JD Edwards, PeopleSoft, SAP i Siebel są przeznaczone wyłącznie do programowania i testowania. Oznacza to, że są one używane tylko podczas programowania i testowania aplikacji.

Po wdrożeniu aplikacji do jej uruchomienia wymagane są licencjonowane adaptory wykonawcze. Jednak podczas budowania usługi adapter można w niej osadzić. Licencja adaptera może pozwalać użyć adaptera osadzonego jako licencjonowanego adaptera wykonawczego. Adaptory te są zgodne z architekturą Java EE Connector Architecture (JCA 1.5). Architektura JCA jest otwartym standardem platformy Java EE zapewniającym łączność z systemem EIS. Architektura JCA udostępnia środowisko zarządzane, co oznacza, że serwer aplikacji obsługuje mechanizm Quality of Service (QoS) zapewniający zarządzanie cyklem życia oraz bezpieczeństwo transakcji. Są one także zgodne ze specyfikacją Enterprise Metadata Discovery z wyjątkiem modułów IBM CICS ECI Resource Adapter i IBM IMS Connector for Java.

Kreator obsługuje także starszy zestaw adapterów, czyli adaptory produktu WebSphere Business Integration.

Zasoby środowiska Java EE

Moduł EIS, czyli moduł SCA zgodny ze wzorcem modułów EIS, można wdrożyć na platformie Java EE.

Wynikiem wdrożenia modułu EIS na platformie Java EE jest gotowa do uruchomienia aplikacja spakowana do postaci pliku EAR i wdrożona na serwerze. Wszystkie artefakty i zasoby platformy Java EE zostają utworzone, a aplikacja jest skonfigurowana i gotowa do uruchomienia.

Właściwości dynamiczne JCA specyfikacji interakcji i specyfikacji połączenia:

Powiązanie EIS może akceptować dane wejściowe dla interfejsów InteractionSpec i ConnectionSpec określone przy użyciu precyzyjnie zdefiniowanego obiektu danych potomnych dołączonego do ładunku. Umożliwia to realizowanie

dynamicznych interakcji żądanie-odpowiedź z adapterem zasobu za pośrednictwem interfejsu InteractionSpec oraz uwierzytelnianie komponentów za pośrednictwem interfejsu ConnectionSpec.

Interfejs javax.cci.InteractionSpec przynosi informacje o sposobie obsługi żądania interakcji z adapterem zasobu. Może również przynosić informacje na temat tego, w jaki sposób uzyskana została interakcja po wykonaniu żądania. Ten sposób dwukierunkowej komunikacji za pośrednictwem interakcji jest czasem nazywany *konwersacjami*.

Powiązanie EIS oczekuje, aby ładunek będący argumentem adaptera zasobu zawierał obiekt danych potomnych o nazwie **properties**. Ten obiekt danych właściwości będzie zawierał pary nazwa-wartość zawierające nazwę właściwości specyfikacji interakcji w określonym formacie. Stosowane są następujące reguły formatowania:

- Nazwy muszą rozpoczynać się przedrostkiem **IS**, po którym powinna występować nazwa właściwości. Na przykład specyfikacja interakcji z właściwością JavaBeans o nazwie **InteractionId** powinna określać nazwę właściwości jako **ISInteractionId**.
- Para nazwa-wartość reprezentuje nazwę i wartość prostego typu właściwości specyfikacji interakcji.

Interfejs przedstawiony w tym przykładzie określa, że danymi wejściowymi operacji jest obiekt danych **Account**. Ten interfejs wywołuje aplikację powiązania importu EIS w celu wysyłania i odbierania dynamicznej właściwości interfejsu InteractionSpec o nazwie **workingSet** i wartości **xyz**.

Graf biznesowy lub obiekty biznesowe na serwerze zawierają bazowy obiekt biznesowy **properties**, który umożliwia wysyłanie danych specyficznych dla protokołu wraz z ładunkiem. Ten obiekt biznesowy **properties** jest wbudowany i nie trzeba go określać w schemacie XML podczas konstruowania obiektu biznesowego. Wymaga on jedynie utworzenia i użycia. Jeśli użytkownik ma własne typy danych zdefiniowane na podstawie schematu XML, należy określić element **properties** zawierający oczekiwane pary nazwa-wartość.

```
BOFactory dataFactory = (BOFactory) \
serviceManager.locateService("com/ibm/websphere/bo/BOFactory");
//Opakowanie dla interfejsów w stylu document/literal wrapped
//przejdźcie do ładunku w przypadku elementów innych niż document/literal
DataObject docLitWrapper = dataFactory.createElement /
("http://mytest/eis/Account", "AccountWrapper");
```

Utworzenie ładunku.

```
DataObject account = docLitWrapper.createDataObject(0);
DataObject accountInfo = account.createDataObject("AccountInfo");
//Konfigurowanie ładunku
```

```
//Konstruowanie danych właściwości na potrzeby interakcji dynamicznej
```

```
DataObject properties = account.createDataObject("properties");
```

Określanie oczekiwanej wartości (**xyz**) dla nazwy workingSet.

```
properties.setString("ISworkingSet", "xyz");
```

```
//Wywołanie usługi z argumentem
```

```
Service accountImport = (Service) \
serviceManager.locateService("AccountOutbound");
DataObject result = accountImport.invoke("createAccount", docLitWrapper);
```

```
//Uzyskanie zwróconej właściwości
DataObject retProperties = result.getDataObject("properties");
```

```
String workingset = retProperties.getString("ISworkingSet");
```

Użytkownik może użyć właściwości interfejsu ConnectionSpec w celu dynamicznego uwierzytelnienia komponentu. Mają zastosowanie te same reguły co powyżej, oprócz tego, że przedrostkiem nazwy właściwości musi być **CS** (zamiast **IS**). Właściwości interfejsu ConnectionSpec nie są dwukierunkowe. Ten sam obiekt danych **properties** może zawierać zarówno właściwości IS, jak i CS.

Aby użyć właściwości interfejsu ConnectionSpec, należy ustawić atrybut **resAuth** określany w powiązaniu importu na wartość **Application**. Należy również upewnić się, czy adapter zasobu obsługuje autoryzację komponentu. Szczegółowe informacje zawiera rozdział 8 specyfikacji J2EE Connector Architecture Specification (specyfikacji architektury konektora J2EE).

Klienty zewnętrzne z powiązaniem EIS:

Serwer może wysyłać komunikaty do klientów zewnętrznych i odbierać komunikaty od nich za pośrednictwem powiązań EIS.

Klient zewnętrzny, na przykład portal WWW lub system EIS, może wymagać wysyłania komunikatów do modułu SCA na serwerze lub może wymagać wywołania przez komponent znajdujący się na serwerze.

Klient wywołuje import EIS w taki sam sposób, jak w przypadku innych aplikacji, przy użyciu interfejsu wywoływania dynamicznego (DII) lub interfejsu Java.

1. Klient zewnętrzny tworzy instancję klasy ServiceManager i wyszukuje import EIS przy użyciu jej nazwy odwołania. Wynikiem wyszukiwania jest implementacja interfejsu usługi.
2. Klient tworzy argument wejściowy - ogólny obiekt danych utworzony dynamicznie przy użyciu schematu obiektu danych. Ten krok jest realizowany przy użyciu implementacji interfejsu DataFactory obiektu SDO.
3. Klient zewnętrzny wywołuje system EIS i uzyskuje wymagane wyniki.

Klient może też wywołać import EIS przy użyciu interfejsu Java.

1. Klient tworzy instancję klasy ServiceManager i wyszukuje import EIS przy użyciu jej nazwy odwołania. Wynikiem wyszukiwania jest interfejs Java importu EIS.
2. Klient tworzy argument wejściowy i obiekt danych określonego typu.
3. Klient wywołuje system EIS i uzyskuje wymagane wyniki.

Interfejs eksportu EIS definiuje interfejs wyeksportowanego komponentu SCA, który jest dostępny dla zewnętrznych aplikacji EIS. Ten interfejs może być postrzegany jako interfejs wywoływany przez aplikację zewnętrzną (na przykład SAP lub PeopleSoft) za pośrednictwem implementacji środowiska wykonawczego aplikacji eksportu EIS.

Eksport używa klasy EISExportBinding do powiązania wyeksportowanych usług z zewnętrzną aplikacją EIS. Umożliwia to subskrybowanie aplikacji zawartej w module SCA w celu nasłuchiwanie żądań usługi EIS. Powiązanie eksportu EIS określa odwzorowanie między definicją zdarzeń przychodzących, w sposób postrzegany przez adapter zasobu (przy użyciu interfejsów architektury Java EE Connector Architecture - JCA), a wywołaniem operacji SCA.

Klasa EISExportBinding wymaga, aby zewnętrzne usługi EIS były oparte na kontraktach przychodzących architektury Java EE Connector Architecture (JCA) 1.5. Klasa EISExportBinding wymaga też, aby na poziomie powiązania lub na poziomie metody określone były procedura obsługi danych lub powiązanie danych.

Powiązania JMS

Dostawca JMS (Java Message Service) umożliwia przesyłanie komunikatów na podstawie interfejsu API i modelu programowania JMS (Java Message Service). Udostępnia on fabryki połączeń JMS, które umożliwiają tworzenie połączeń dla miejsc docelowych JMS oraz wysyłanie i odbieranie komunikatów.

Powiązania JMS można użyć w interakcjach z powiązaniem dostawcy magistrali integracji usług SIB (Service Integration Bus). Są one zgodne ze specyfikacjami JMS i JCA 1.5.

Powiązania eksportu i importu JMS umożliwiają modułowi SCA (Service Component Architecture) wywoływanie zewnętrznych systemów JMS i odbieranie komunikatów z tych systemów.

Powiązania importu i eksportu JMS udostępniają integrację z aplikacjami JMS za pośrednictwem dostawcy JMS magistrali SIB opartego na architekturze JCA 1.5, który stanowi część serwera WebSphere Application Server. Inne adaptory zasobów JMS oparte na architekturze JCA 1.5 nie są obsługiwane.

Przegląd powiązań JMS:

Powiązania JMS udostępniają połączenia między środowiskiem architektury SCA (Service Component Architecture) i systemami JMS.

Powiązania JMS

Główne komponenty powiązań importu JMS i powiązań eksportu JMS są następujące:

- Adapter zasobów: umożliwia zarządzane, dwukierunkowe połączenia między modułem SCA i zewnętrznymi systemami JMS.
- Połączenia: hermetyzują wirtualne połączenie między klientem i aplikacją dostawcy.
- Miejsca docelowe: używane przez klienta do określenia celu tworzonych komunikatów lub źródła konsumowanych komunikatów.
- Dane uwierzytelniania: używane w celu zabezpieczenia dostępu do powiązania.

Kluczowe opcje powiązań usług JMS

Nagłówki specjalne

W elementach importowanych i eksportowanych JMS stosowane są specjalne właściwości nagłówka zawierające informacje pozwalające elementowi docelowemu na właściwą obsługę komunikatu.

Na przykład właściwość `TargetFunctionName` odwzorowuje metodę własną w metodę operacji.

Zasoby środowiska Java EE

Podczas wdrażania elementów importowanych i eksportowanych JMS w środowisku Java EE utworzona zostaje pewna liczba zasobów środowiska Java EE.

Fabryka połączeń

Używana przez klienty do tworzenia połączenia z dostawcą usług JMS.

Specyfikacja aktywacji

Używana przez elementy importowane do odbierania odpowiedzi na żądanie, a przez elementy eksportowane podczas konfigurowania punktów końcowych komunikatów reprezentujących procesy nasłuchiwanie komunikatów w ich interakcjach z systemem przesyłania komunikatów.

Miejsca docelowe

- Miejsce docelowe wysyłania: w elemencie importowanym jest to miejsce, do którego jest wysyłane żądanie lub komunikat wychodzący; w elemencie eksportowanym jest to miejsce, do którego jest wysyłany komunikat odpowiedzi, jeśli nie jest zastąpione za pomocą pola nagłówka `JMSReplyTo` w komunikacie przychodzącym.
- Miejsce docelowe odbioru: jest to miejsce, w którym należy umieścić komunikat przychodzący. W przypadku elementów importowanych jest to odpowiedź, a w przypadku elementów eksportowanych - żądanie.
- Miejsce docelowe wywołania zwrotnego: jest to miejsce docelowe w systemie JMS architektury SCA używane do zapisywania informacji o korelacji. Tego miejsca docelowego nie należy odczytywać ani nie należy do niego zapisywać.

Czynność instalacji powoduje utworzenie zasobu fabryki połączeń oraz trzech miejsc docelowych. Jej wynikiem jest także utworzenie zasobu specyfikacji aktywowania umożliwiającego procesowi nasłuchiwanie komunikatów czasu wykonywania nasłuchiwanie odpowiedzi w miejscu docelowym odbioru. Właściwości tych zasobów są określone w pliku importu lub eksportu.

Integracja JMS i adaptery zasobów:

Interfejs JMS (Java Message Service) udostępnia integrację za pośrednictwem dostępnego adaptera zasobu JMS opartego na architekturze JCA 1.5. Dla adaptera zasobu JMS magistrali integracji usług (Service Integration Bus - SIB) udostępniana jest pełna obsługa integracji JMS.

Aby przeprowadzić integrację z zewnętrznym systemem JMS zgodnym ze standardem JCA 1.5, należy użyć adaptera zasobu dostawcy JMS dla architektury JCA 1.5. Usługi zewnętrzne zgodne ze standardem JCA 1.5 mogą odbierać komunikaty i wysyłać komunikaty w celu integracji z komponentami SCA (Service Component Architecture) przy użyciu adaptera zasobu JMS magistrali SIB.

Używanie innych adapterów zasobów architektury JCA 1.5 pochodzących od konkretnych dostawców nie jest obsługiwane.

Powiązania importu i eksportu JMS:

Powiązania importu i eksportu JMS umożliwiają interakcje modułów SCA z usługami udostępnianymi przez zewnętrzne aplikacje JMS.

Powiązania importu JMS

Połączenia z powiązaniem dostawcą JMS miejsc docelowych JMS są tworzone przy użyciu fabryki połączeń JMS. Do zarządzania fabrykami połączeń JMS dla domyślnego dostawcy przesyłania komunikatów należy użyć obiektów administracyjnych fabryki połączeń.

Interakcja z zewnętrznymi systemami JMS obejmuje używanie miejsc docelowych do wysyłania żądań i odbierania odpowiedzi.

Obsługiwane są dwa typy scenariuszy używania dla powiązania importu JMS, zależne od typu wywoływanej operacji:

- **Jednokierunkowy:** import JMS umieszcza komunikat w miejscu docelowym wysyłania skonfigurowanym w powiązaniu importu. Pole `replyTo` nagłówka JMS nie jest ustawiane.
- **Dwukierunkowy (żądanie-odpowiedź):** import JMS umieszcza komunikat w miejscu docelowym wysyłania, a następnie utrzuca odpowiedź otrzymaną z komponentu SCA.

Powiązanie importu może być skonfigurowane (przy użyciu pola **Schemat korelacji odpowiedzi** w produkcie Integration Designer) tak, aby oczekiwać skopiowania identyfikatora korelacji komunikatu odpowiedzi z identyfikatora komunikatu żądania (wartość domyślna) lub z identyfikatora korelacji komunikatu żądania. Powiązanie importu może również być skonfigurowane tak, aby używać tymczasowego, dynamicznego miejsca docelowego odpowiedzi w celu korelowania odpowiedzi z żądaniami. Tymczasowe miejsce docelowe jest tworzone dla każdego żądania, a import używa tego miejsca docelowego do odbierania odpowiedzi.

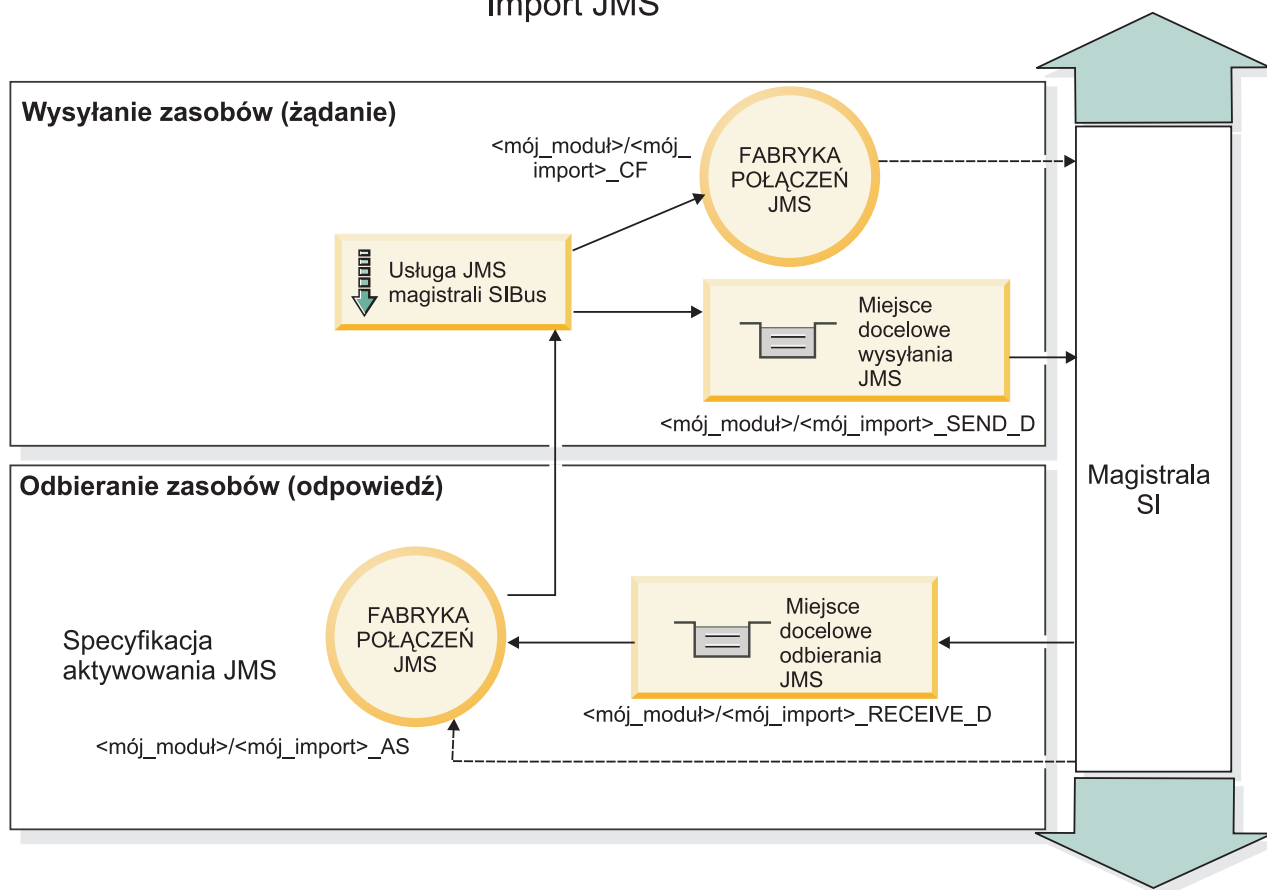
Miejsce docelowe `receive` (odbioru) jest ustawiane we właściwości nagłówka `replyTo` komunikatu wychodzącego. Wdrażany jest proces nasłuchiwanie komunikatów, który nasłuchuje w miejscu docelowym odpowiedzi, a po otrzymaniu odpowiedzi ten proces nasłuchiwanie komunikatów przekazuje odpowiedź z powrotem do komponentu.

Istnieje możliwość określenia dynamicznych i statycznych właściwości nagłówka zarówno dla jednokierunkowych, jak i dwukierunkowych scenariuszy używania. Właściwości statyczne można ustawić z poziomu powiązania metody importu JMS. Niektóre z tych właściwości mają specjalne znaczenie dla środowiska wykonawczego JMS architektury SCA.

Należy pamiętać, że powiązanie JMS jest powiązaniem asynchronicznym. Jeśli komponent wywołujący wywołuje import JMS w sposób synchroniczny (dla operacji dwukierunkowej), komponent wywołujący jest blokowany do chwili zwrócenia odpowiedzi przez usługę JMS.

Na ilustracji Rys. 32 na stronie 118 przedstawiono sposób połączenia importu z usługą zewnętrzną.

Import JMS



Rysunek 77. Zasoby powiązania importu JMS

Powiązania eksportu JMS

Powiązania eksportu JMS zapewniają modułom SCA sposób na udostępnianie usług zewnętrznym aplikacjom JMS.

Powiązanie będące częścią eksportu JMS jest specyfikacją aktywowania, którą można konfigurować.

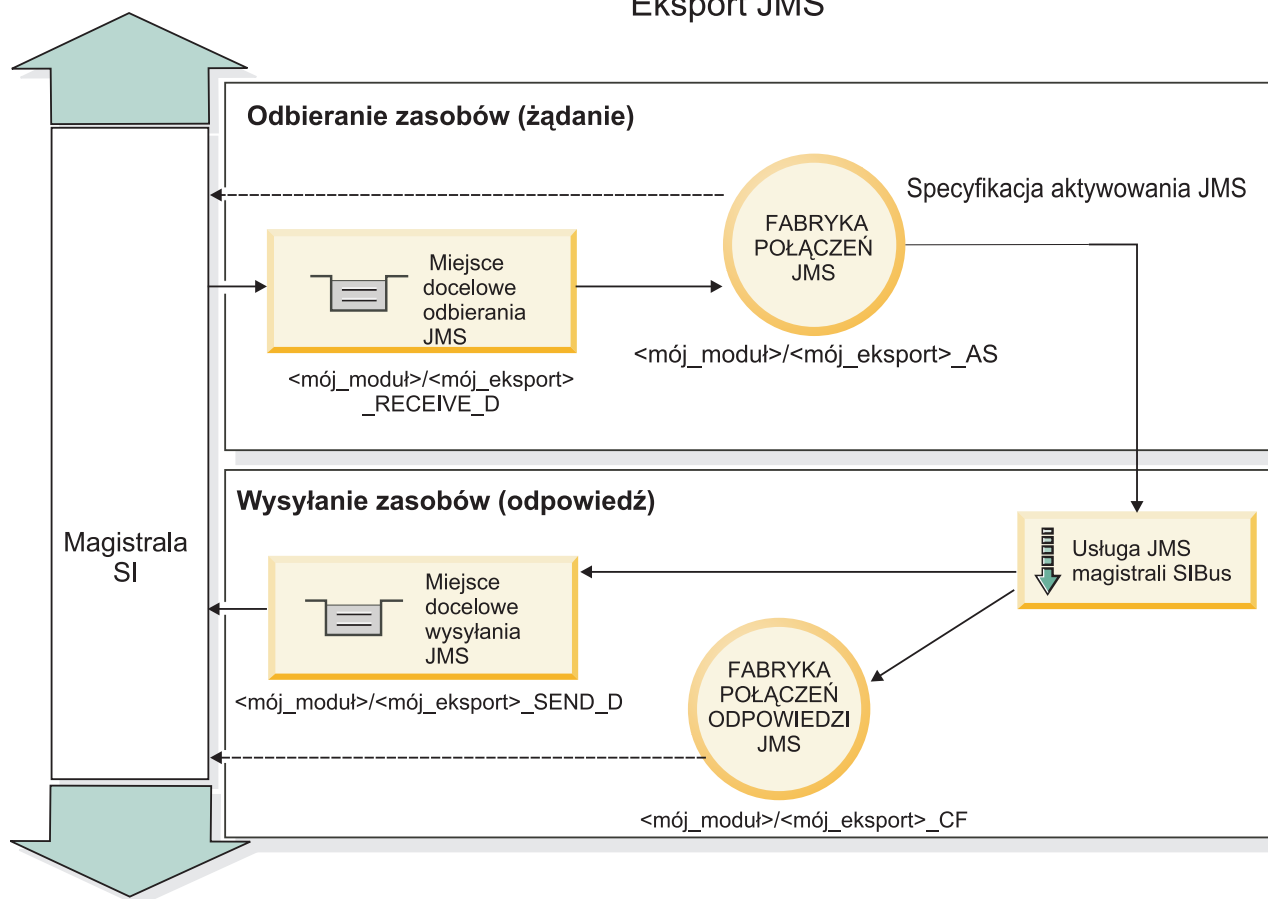
Eksport JMS ma miejsca docelowe wysyłania i odbierania.

- Miejsce docelowe receive (odbioru) jest miejscem, w którym powinny być umieszczane komunikaty przychodzące do komponentu docelowego.
- Miejsce docelowe send (wysyłania) jest miejscem, do którego wysyłana będzie odpowiedź, chyba że komunikat przychodzący nadpisuje to miejsce przy użyciu właściwości nagłówka replyTo.

Wdrażany jest proces nasłuchiwanie komunikatów, który nasłuchuje żądań przychodzących do miejsca docelowego odbierania określonego w powiązaniu eksportu. Miejsce docelowe określone w polu send służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywoływany komponent udostępnia odpowiedź. Miejsce docelowe określone w polu replyTo komunikatu przychodzącego nadpisuje miejsce docelowe określone w polu send.

Na ilustracji Rys. 33 na stronie 119 przedstawiono sposób połączenia zewnętrznego requestera z eksportem.

Eksport JMS



Rysunek 78. Zasoby powiązania eksportu JMS

Nagłówki JMS:

Komunikat JMS zawiera dwa typy nagłówków: nagłówek systemowy JMS i wiele właściwości JMS. Dostęp do obu typów nagłówków można uzyskać w module mediacji, w obiekcie SMO lub przy użyciu funkcji API ContextService.

Nagłówek systemu JMS

Nagłówek systemu JMS jest reprezentowany w obiekcie SMO przez element JMSHeader, który zawiera wszystkie pola znajdujące się zazwyczaj w nagłówku JMS. Mimo że mogą one być modyfikowane podczas mediacji (lub w funkcji ContextService), niektóre pola nagłówka systemu JMS ustawione w obiekcie SMO nie będą propagowane w wychodzącym komunikacie JMS, ponieważ są nadpisywane przez wartości systemowe lub statyczne.

Kluczowymi polami w nagłówku systemu JMS, które mogą być aktualizowane podczas mediacji (lub w funkcji ContextService), są:

- **JMSType** i **JMSCorrelationID** – wartości konkretnych właściwości predefiniowanego nagłówka komunikatu.
- **JMSDeliveryMode** – wartości dla trybu dostarczania (persistent - trwały lub nonpersistent - nietrwały; wartość domyślna to persistent).
- **JMSPriority** – wartość priorytetu (od 0 do 9, wartość domyślna to JMS_Default_Priority).

Właściwości JMS

Właściwości JMS są reprezentowane w obiekcie SMO jako pozycje na liście Properties (właściwości). Właściwości można dodawać, aktualizować lub usuwać podczas mediacji lub przy użyciu funkcji API ContextService.

Właściwości mogą być też ustawiane statycznie w powiązaniu JMS. Właściwości ustawione statycznie nadpisują ustawienia (o tej samej nazwie), które są ustawione dynamicznie.

Właściwości użytkownika propagowane z innych powiązań (na przykład powiązania HTTP) będą wyprowadzane w powiązaniu JMS jako właściwości JMS.

Ustawienia propagowania nagłówka

Propagowanie nagłówka systemu JMS i właściwości z przychodzącego komunikatu JMS do kolejnych komponentów lub z wcześniejszych komponentów do wychodzącego komunikatu JMS może być kontrolowane za pośrednictwem flagi Propaguj nagłówki protokołu w powiązaniu.

Jeśli flaga Propaguj nagłówki protokołu jest ustawiona, informacje w nagłówku mogą przepływać do komunikatu lub do komponentu docelowego, jak opisano poniżej:

- **Żądanie eksportu JMS**

Nagłówek JMS odebrany w komunikacie będzie propagowany do komponentów docelowych za pośrednictwem usługi kontekstu. Właściwości JMS odebrane w komunikacie będą propagowane do komponentów docelowych za pośrednictwem usługi kontekstu.

- **Odpowiedź eksportu JMS**

Wszystkie pola nagłówka JMS ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu eksportu JMS. Wszystkie właściwości ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu eksportu JMS.

- **Żądanie importu JMS**

Wszystkie pola nagłówka JMS ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu importu JMS. Wszystkie właściwości ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu importu JMS.

- **Odpowiedź importu JMS**

Nagłówek JMS odebrany w komunikacie będzie propagowany do komponentów docelowych za pośrednictwem usługi kontekstu. Właściwości JMS odebrane w komunikacie będą propagowane do komponentów docelowych za pośrednictwem usługi kontekstu.

Schemat korelacji tymczasowego, dynamicznego miejsca docelowego odpowiedzi JMS:

Schemat korelacji tymczasowego, dynamicznego miejsca docelowego odpowiedzi powoduje utworzenie unikalnej dynamicznej kolejki lub dynamicznego tematu dla każdego wysłanego żądania.

Do określenia rodzaju kolejki lub tematu tymczasowego, dynamicznego miejsca docelowego używane jest statyczne miejsce docelowe odpowiedzi określone w imporcie. Jest ono ustawiane w polu **ReplyTo** żądania, a import JMS nasłuchuje odpowiedzi w tym miejscu docelowym. Po odebraniu odpowiedzi jest ona ponownie umieszczana w kolejce statycznego miejsca docelowego odpowiedzi w celu przetworzenia asynchronicznego. Pole **CorrelationID** odpowiedzi nie jest używane i nie trzeba go ustawiać.

Problemy związane z transakcjami

Jeśli używane jest tymczasowe, dynamiczne miejsce docelowe, odpowiedź musi być konsumowana w tym samym wątku co wysłana odpowiedź. Żądanie musi być wysłane poza globalną transakcją i musi być zatwierdzone, zanim zostanie odebrane przez usługę zaplecza i zanim zostanie zwrócona odpowiedź.

Trwałość

Tymczasowe kolejki dynamiczne są jednostkami krótkotrwałymi i nie gwarantują tego samego poziomu trwałości, co statyczna kolejka lub statyczny temat. Tymczasowa kolejka dynamiczna lub tymczasowy temat dynamiczny nie

przetwarzają restartu serwera (podobnie jak komunikaty). Po ponownym umieszczeniu komunikatu w kolejce do statycznego miejsca docelowego odpowiedzi, komunikat ten zachowuje zdefiniowaną w nim trwałość.

Limit czasu

Import oczekuje odpowiedzi w tymczasowym, dynamicznym miejscu docelowym odpowiedzi przez ustalony czas. Ten okres jest pobierany z kwalifikatora czasu ważności odpowiedzi architektury SCA, jeśli jest on ustawiony. W przeciwnym razie wartością domyślną tego okresu jest 60 sekund. Jeśli czas oczekiwania zostanie przekroczony, import zgłasza wyjątek `ServiceTimeoutRuntimeException`.

Klienty zewnętrzne:

Serwer może wysyłać komunikaty do klientów zewnętrznych i odbierać komunikaty od nich za pośrednictwem powiązań JMS.

Klient zewnętrzny (na przykład portal WWW lub system EIS) może wysyłać komunikat do modułu SCA na serwerze lub może być wywoływany przez komponent znajdujący się na serwerze.

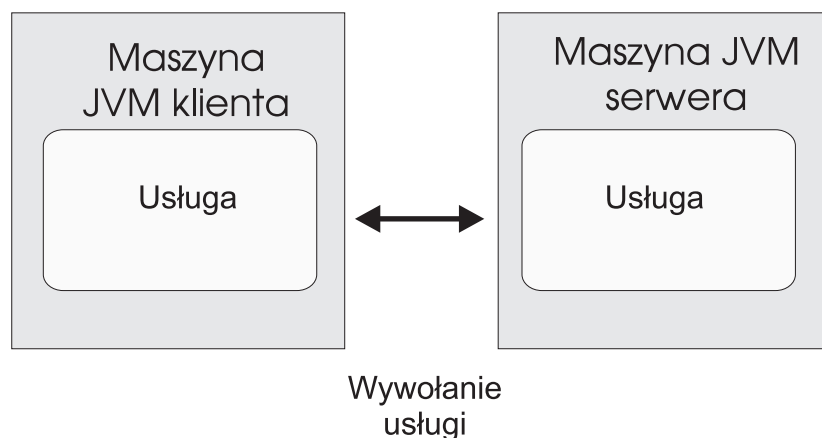
Komponenty eksportu JMS wdrażają procesy nasłuchiwanie komunikatów w celu nasłuchiwanie żądań przychodzących do miejsca docelowego odbierania określonego w powiązaniu eksportu. Miejsce docelowe określone w polu `send` służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołana aplikacja udostępnia odpowiedź. Dzięki temu klient zewnętrzny może wywoływać aplikacje przy użyciu powiązania eksportu.

Importy JMS wchodzi w interakcje z klientami zewnętrznymi, wysyłając komunikaty do kolejek JMS i otrzymując z nich komunikaty.

Praca z klientami zewnętrznymi:

Klient zewnętrzny (znajdujący się poza serwerem) może wymagać interakcji z aplikacją zainstalowaną na serwerze.

Poniżej rozważany jest prosty scenariusz, w którym klient zewnętrzny ma współpracować z pewną aplikacją na serwerze. Na rysunku przedstawiono typowy prosty scenariusz.



Rysunek 79. Scenariusz prostego przypadku użycia: klient zewnętrzny realizuje interakcję z aplikacją serwera

Aplikacja SCA obejmuje element eksportowany z powiązaniem JMS. Dzięki temu aplikacja jest dostępna dla klientów zewnętrznych.

Gdy istnieje klient zewnętrzny w wirtualnej maszynie języka Java (JVM), w której nie działa serwer, należy wykonać kilka kroków, aby utworzyć połączenie i zrealizować interakcję z elementem eksportowanym JMS. Klient uzyskuje kontekst początkowy z poprawnymi wartościami i wyszukuje zasoby za pośrednictwem interfejsu JNDI. Następnie

klient używa klienta zgodnego ze specyfikacją JMS 1.1 w celu uzyskania dostępu do miejsc docelowych oraz komunikatów o wysyłaniu i odbieraniu w miejscach docelowych.

Domyślne nazwy JNDI zasobów tworzonych automatycznie przez środowisko wykonawcze są wymienione w tej sekcji w temacie dotyczącym konfiguracji. Jeśli jednak zasoby zostały utworzone wcześniej, należy używać tych nazw JNDI.

1. Skonfiguruj miejsca docelowe JMS oraz fabrykę połączeń w celu wysłania komunikatu.
2. Upewnij się o poprawności kontekstu JNDI, portu adaptera zasobów SIB oraz portu startowego przesyłania komunikatów.
Serwer używa pewnych portów domyślnych, ale gdy w danym systemie zainstalowanych jest więcej serwerów, podczas instalacji tworzone są porty alternatywne, aby uniknąć konfliktów z innymi instancjami serwerów. Do określania portów używanych przez serwer można używać Konsoli administracyjnej. Należy przejść do opcji **Serwery > Serwery aplikacji > nazwa_używanego_serwera > Konfiguracja** i kliknąć opcję **Porty** w sekcji **Komunikacja**. Następnie można przeprowadzić edycję używanego portu.
3. Klient uzyskuje kontekst początkowy z poprawnymi wartościami i wyszukuje zasoby za pośrednictwem interfejsu JNDI.
4. Za pomocą specyfikacji JMS 1.1 klient uzyskuje dostęp do miejsc docelowych oraz komunikatów o wysyłaniu i odbieraniu w miejscach docelowych.

Rozwiązywanie problemów z powiązaniem JMS:

Istnieje możliwość diagnozowania i rozwiązywania problemów związanych z powiązaniem JMS.

Wyjątki implementacji

W odpowiedzi na różne warunki błędu implementacja elementu importowanego lub eksportowanego JMS może zwracać wyjątki dwóch typów:

- Wyjątek biznesowy usługi: ten wyjątek jest zwracany po wystąpieniu uszkodzenia określonego w interfejsie biznesowym usługi (typ portu WSDL).
- Wyjątek środowiska wykonawczego usługi: jest zgłaszany we wszystkich innych przypadkach. Najczęściej wyjątek *cause* (przyczyna) będzie zawierać pierwotny wyjątek (*JMSEException*).

Element importowany może na przykład oczekiwać tylko jednego komunikatu odpowiedzi na każdy komunikat żądania. Jeśli nadejdzie więcej niż jedna odpowiedź lub nadejdzie odpowiedź spóźniona (czas utraty ważności odpowiedzi SCA minął), zgłoszony zostanie wyjątek środowiska wykonawczego usługi. Transakcja zostanie wycofana, a komunikat odpowiedzi zostanie usunięty z kolejki lub obsłużony przez menedżer zdarzeń zakończonych niepowodzeniem.

Podstawowe warunki niepowodzenia

Podstawowe warunki niepowodzenia powiązań JMS są określane przez semantykę transakcyjną, przez konfigurację dostawcy JMS lub przez odwołanie do aktualnego działania innych komponentów. Istnieją następujące podstawowe warunki niepowodzenia:

- Niepowodzenie nawiązania połączenia z dostawcą JMS lub miejscem docelowym.
Niepowodzenie nawiązania połączenia z dostawcą JMS w celu odbierania komunikatów powoduje, że nie można uruchomić procesu nasłuchiwanie komunikatów. Ten warunek zostaje zarejestrowany w dzienniku produktu WebSphere Application Server. Trwałe komunikaty pozostaną w miejscu docelowym do czasu ich pomyślnego pobrania (lub do utraty ważności).
Niepowodzenie nawiązania połączenia z dostawcą JMS w celu wysyłania komunikatów wychodzących powoduje wycofanie transakcji sterującej wysyłaniem.
- Niepowodzenie analizy komunikatu przychodzącego lub budowy komunikatu wychodzącego.
Niepowodzenie w powiązaniu danych lub procedurze obsługi danych powoduje wycofanie transakcji sterującej wykonywanym działaniem.
- Niepowodzenie wysłania komunikatu wychodzącego.

- Niepowodzenie wysłania komunikatu powoduje wycofanie odpowiedniej transakcji.
- Wiele komunikatów odpowiedzi lub nieoczekiwane, spóźnione komunikaty odpowiedzi.

Element importowany oczekuje tylko jednego komunikatu odpowiedzi na każdy komunikat żądania. Ponadto poprawny przedział czasu, w którym można odebrać odpowiedź, jest określany przez kwalifikator utraty ważności odpowiedzi SCA znajdujący się w żądaniu. Po nadejściu odpowiedzi lub upływie czasu utraty ważności rekord korelacji zostaje usunięty. W przypadku nadejścia nieoczekiwanej lub spóźnionej odpowiedzi zostaje zgłoszony wyjątek środowiska wykonawczego usługi.
- Wyjątek środowiska wykonawczego związany z przekroczeniem limitu czasu usługi spowodowany przez spóźnioną odpowiedź, gdy używany jest tymczasowy, dynamiczny schemat korelacji miejsca docelowego odpowiedzi.

Przekroczenie limitu czasu elementu importowanego JMS następuje po upływie czasu określonego przez kwalifikator utraty ważności odpowiedzi SCA (gdy nie jest on ustawiony, domyślnie przyjmuje się czas 60 sekund).

Komunikaty SCA oparte na usługach JMS nie są wyświetlane w menedżerze zdarzeń zakończonych niepowodzeniem

Jeśli komunikaty SCA są wynikiem niepowodzenia interakcji JMS, powinny znaleźć się w menedżerze zdarzeń zakończonych niepowodzeniem. Jeśli takie komunikaty nie pojawiają się w menedżerze zdarzeń zakończonych niepowodzeniem, należy upewnić się, że w bazowym miejscu docelowym SIB miejsca docelowego JMS maksymalną liczbę nieudanych dostaw ustawiono na wartość większą niż 1. Ustawienie wartości 2 lub większej umożliwia interakcję z menedżerem zdarzeń zakończonych niepowodzeniem w trakcie wywołań SCA dotyczących powiązań JMS.

Obsługa wyjątków:

Sposób skonfigurowania powiązania określa sposób obsługi wyjątków wywołanych przez procedury obsługi danych lub powiązania danych. Na działanie systemu podczas zgłoszenia takiego wyjątku dodatkowy wpływ ma rodzaj przepływu mediacji.

Jeśli procedura obsługi danych lub powiązanie danych zostanie wywołane przez powiązanie użytkownika, może wystąpić wiele różnych problemów. Procedura obsługi danych może na przykład odebrać komunikat z uszkodzonym ładunkiem lub może próbować odczytać komunikat w niepoprawnym formacie.

Sposób obsługi takiego wyjątku przez powiązanie użytkownika jest określany na podstawie tego, w jaki sposób zaimplementowano procedurę obsługi danych lub powiązanie danych. Zalecanym sposobem utworzenia powiązania danych jest zaprojektowanie go tak, aby zgłaszało wyjątek **DataBindingException**.

W przypadku wywołania wyjątku czasu wykonywania, między innymi wyjątku **DataBindingException**, mają miejsce następujące działania:

- Jeśli przepływ mediacji jest skonfigurowany jako transakcyjny, komunikat JMS jest domyślnie zapisywany w menedżerze zdarzeń zakończonych niepowodzeniem w celu ręcznego powtórzenia lub usunięcia.

Uwaga: Użytkownik może zmienić tryb odtwarzania dla powiązania tak, aby komunikat był wycofywany zamiast zapisywania go w menedżerze zdarzeń zakończonych niepowodzeniem.

- Jeśli przepływ mediacji nie jest transakcyjny, wyjątek jest rejestrowany, a komunikat zostaje utracony.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, każdy wyjątek procedury obsługi danych jest opakowany w wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

Ogólne powiązania JMS

Ogólne powiązanie JMS udostępnia połączenia z dostawcami innych firm zgodnymi ze standardem JMS 1.1. Ogólne powiązanie JMS działa w sposób podobny do powiązań JMS.

Usługa udostępniana za pośrednictwem powiązania JMS umożliwia modułowi SCA (Service Component Architecture) wywoływanie systemów zewnętrznych lub odbieranie komunikatów z tych systemów. System może być zewnętrznym systemem JMS.

Ogólne powiązanie JMS umożliwia integrację z dostawcami JMS, które nie są zgodne z architekturą JCA 1.5, ale obsługują interfejs JMS 1.1 i implementują opcjonalne narzędzie serwera aplikacji JMS. Ogólne powiązanie JMS obsługuje dostawców JMS (w tym Oracle AQ, TIBCO, SonicMQ, WebMethods oraz BEA WebLogic) nieobsługujących specyfikacji JCA 1.5, ale obsługujących narzędzie serwera aplikacji (Application Server Facility) specyfikacji JMS 1.1. Wbudowany dostawca JMS produktu WebSphere (SIBJMS), który jest dostawcą JMS opartym na architekturze JCA 1.5, nie jest obsługiwany przez to powiązanie. W przypadku używania tego dostawcy należy użyć powiązań omówionych w sekcji "Powiązania JMS" na stronie 116.

Tego powiązania ogólnego należy używać w środowisku SCA podczas przeprowadzania integracji z systemem opartym na interfejsie JMS, który nie jest zgodny z architekturą JCA 1.5. Po jego zastosowaniu docelowe aplikacje zewnętrzne mogą otrzymywać i wysyłać komunikaty w celu integracji z komponentem SCA.

Przegląd ogólnych powiązań JMS:

Ogólne powiązania JMS są powiązaniem JMS, które nie są oparte na architekturze JCA i które udostępniają połączenia między środowiskiem SCA (Service Component Architecture) oraz systemami JMS zgodnymi ze standardem 1.1 i implementującymi opcjonalne narzędzie serwera aplikacji JMS.

Ogólne powiązania JMS

Do podstawowych cech ogólnych powiązań importu i eksportu JMS należą następujące:

- Port nasłuchiwania: umożliwia dostawcom JMS, które nie są oparte na architekturze JCA, odbieranie komunikatów i rozsyłanie ich do komponentu bean sterowanego komunikatami (Message Driven Bean - MDB).
- Połączenia: hermetyzują wirtualne połączenie między klientem i aplikacją dostawcy.
- Miejsca docelowe: używane przez klienta do określenia celu tworzonych komunikatów lub źródła konsumowanych komunikatów.
- Dane uwierzytelniania: używane w celu zabezpieczenia dostępu do powiązania.

Ogólne powiązanie importu JMS

Ogólne powiązanie importu JMS pozwalają komponentom w module SCA komunikować się z usługami udostępnianymi przez zewnętrznych dostawców JMS, które nie są zgodne z architekturą JCA 1.5.

Częścią odpowiedzialną za połączenie w imporcie JMS jest fabryka połączeń. Fabryka połączeń, czyli obiekt używany przez klienta do tworzenia połączenia z dostawcą, hermetyzuje zbiór parametrów konfiguracyjnych połączenia zdefiniowanych przez administratora. Każda fabryka połączeń jest instancją interfejsu ConnectionFactory, QueueConnectionFactory lub TopicConnectionFactory.

Interakcja z zewnętrznymi systemami JMS obejmuje używanie miejsc docelowych do wysyłania żądań i odbierania odpowiedzi.

Obsługiwane są dwa typy scenariuszy używania dla ogólnego powiązania importu JMS, zależne od typu wywoływanej operacji:

- Jednokierunkowy: ogólny import JMS umieszcza komunikat w miejscu docelowym wysyłania skonfigurowanym w powiązaniu importu. Do pola replyTo nagłówek JMS nie są wysyłane żadne informacje.
- Dwukierunkowy (żądanie-odpowiedź): ogólny import JMS umieszcza komunikat w miejscu docelowym wysyłania, a następnie utrwała odpowiedź otrzymaną z komponentu SCA.

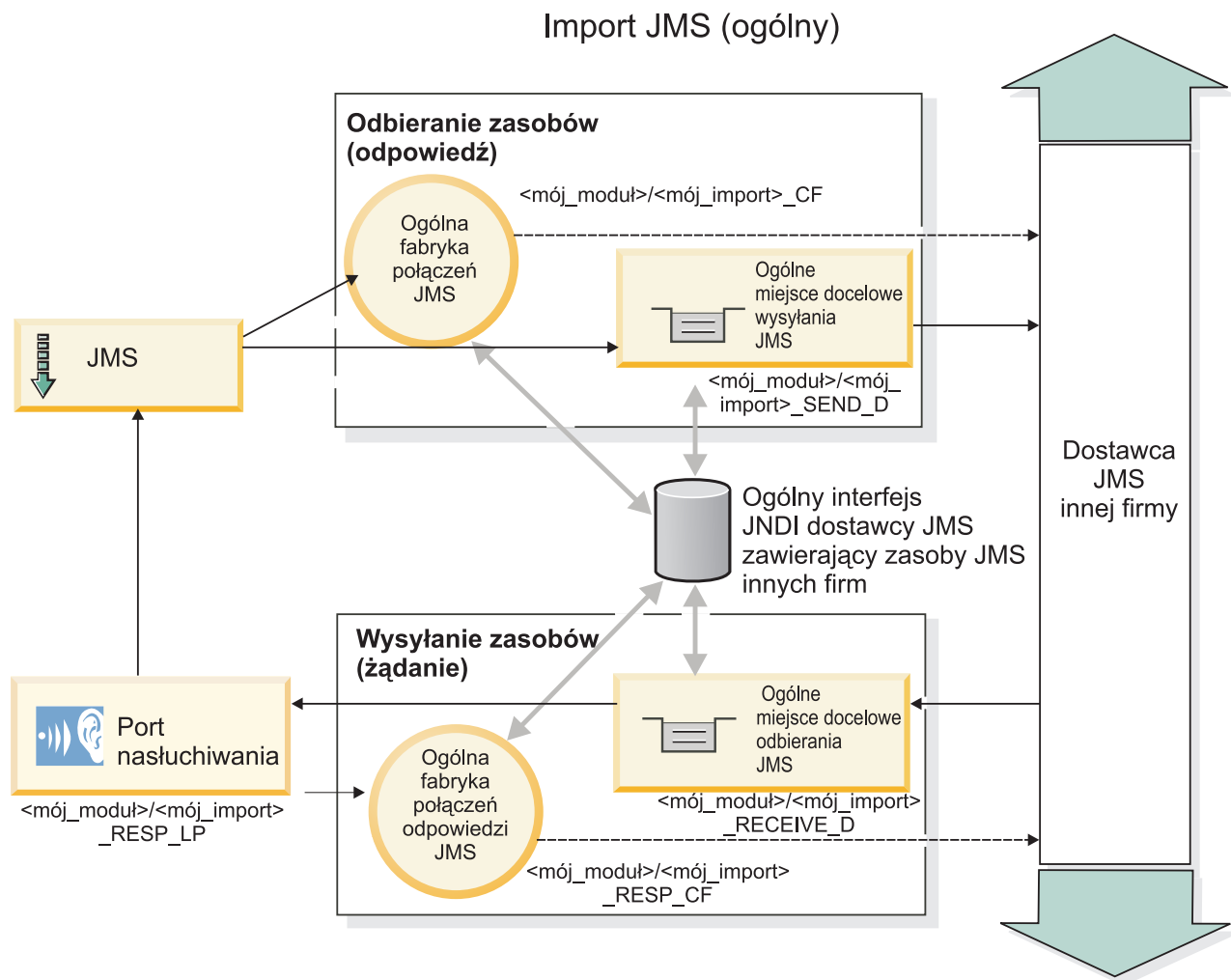
Miejsce docelowe receive (odbioru) jest ustawiane we właściwości nagłówka replyTo komunikatu wychodzącego. Wdrażany jest komponent bean sterowany komunikatami (Message Driven Bean - MDB), który nasłuchuje w miejscu docelowym odbioru, a po otrzymaniu odpowiedzi ten komponent MDB przekazuje odpowiedź z powrotem do komponentu.

Powiązanie importu może być skonfigurowane (przy użyciu pola **Schemat korelacji odpowiedzi** w produkcie Integration Designer) tak, aby oczekiwać skopiowania identyfikatora korelacji komunikatu odpowiedzi z identyfikatora komunikatu żądania (jest to ustawienie domyślne) lub z identyfikatora korelacji komunikatu żądania.

Istnieje możliwość określenia dynamicznych i statycznych właściwości nagłówka zarówno dla jednokierunkowych, jak i dwukierunkowych scenariuszy używania. Właściwości statyczne można ustawić z poziomu powiązania metody ogólnego importu JMS. Niektóre z tych właściwości mają specjalne znaczenie dla środowiska wykonawczego JMS architektury SCA.

Należy pamiętać, że ogólne powiązanie JMS jest powiązaniem asynchronicznym. Jeśli komponent wywołujący wywołuje ogólny import JMS w sposób synchroniczny (dla operacji dwukierunkowej), komponent wywołujący jest blokowany do chwili zwrócenia odpowiedzi przez usługę JMS.

Na ilustracji Rys. 35 na stronie 126 przedstawiono sposób połączenia importu z usługą zewnętrzną.



Rysunek 80. Zasoby ogólnego powiązania importu JMS

Ogólne powiązania eksportu JMS

Ogólne powiązania eksportu JMS zapewniają modułom SCA sposób na udostępnianie usług zewnętrznym aplikacjom JMS.

Część odpowiedzialna za połączenie w eksporcie JMS składa się z fabryki połączeń i portu nasłuchiwania.

Ogólny eksport JMS ma miejsca docelowe wysyłania i odbioru.

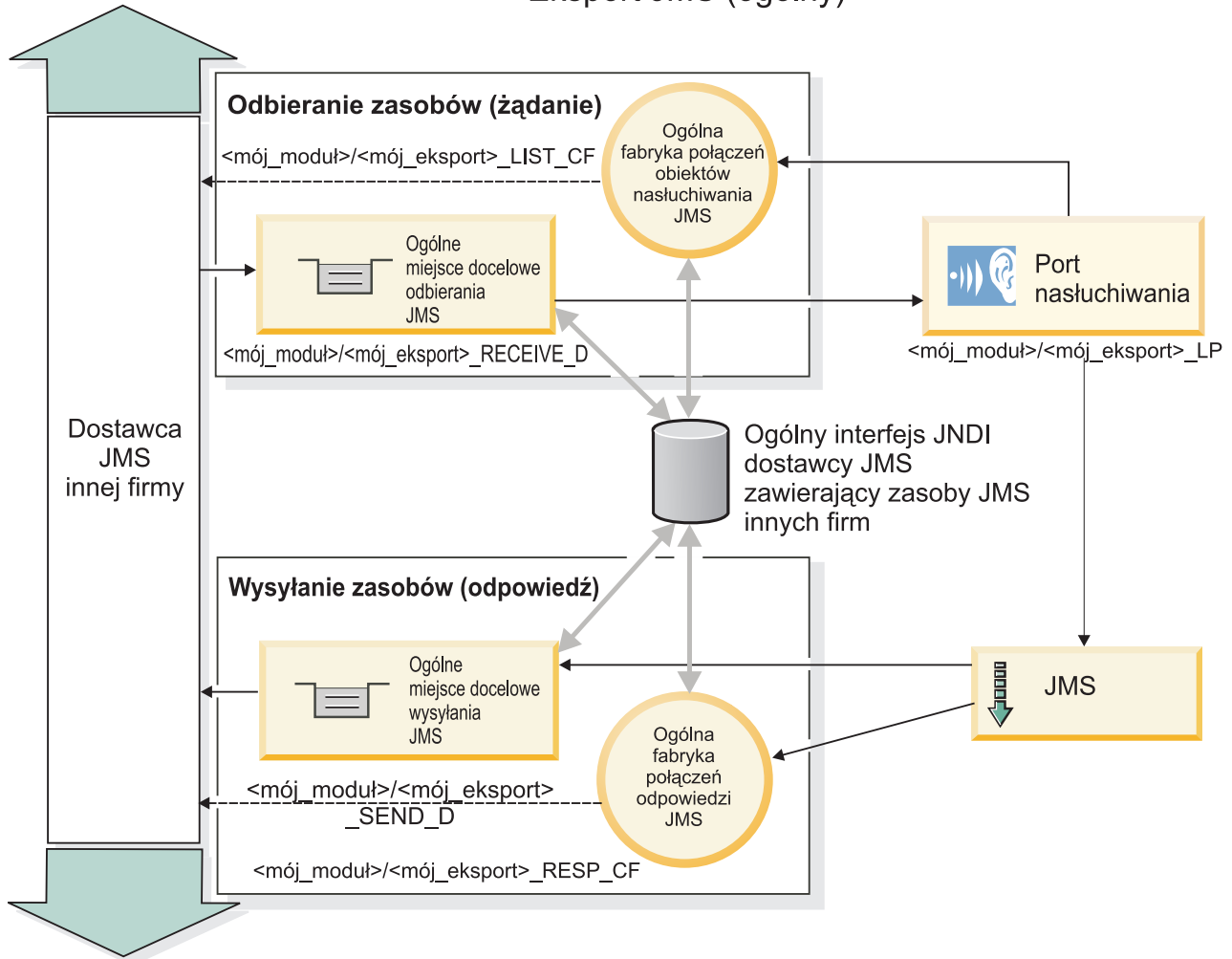
- Miejsce docelowe receive (odbioru) jest miejscem, w którym powinny być umieszczane komunikaty przychodzące do komponentu docelowego.
- Miejsce docelowe send (wysyłania) jest miejscem, do którego wysyłana będzie odpowiedź, chyba że komunikat przychodzący nadpisuje to miejsce przy użyciu właściwości nagłówka replyTo.

Wdrażany jest komponent MDB, który nasłuchuje żądań przychodzących do miejsca docelowego receive (odbioru) określonego w powiązaniu eksportu.

- Miejsce docelowe określone w polu send służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołany komponent udostępnia odpowiedź.
- Miejsce docelowe określone w polu replyTo komunikatu przychodzącego nadpisuje miejsce docelowe określone w polu send.
- W przypadku scenariuszy żądania-odpowiedzi powiązanie importu może być skonfigurowane (przy użyciu pola **Schemat korelacji odpowiedzi** w produkcie Integration Designer) tak, aby oczekiwano że odpowiedź skopiuje wartość pola message ID żądania do pola correlation ID komunikatu odpowiedzi (jest to ustawienie domyślne) lub tak, aby oczekiwano że odpowiedź skopiuje wartość pola correlation ID żądania do pola correlation ID komunikatu odpowiedzi.

Na ilustracji Rys. 36 na stronie 127 przedstawiono sposób połączenia zewnętrznego requestera z eksportem.

Eksport JMS (ogólny)



Rysunek 81. Zasoby ogólnego powiązania eksportu JMS

Kluczowe opcje ogólnych powiązań usług JMS:

Opcje ogólnych powiązań elementów importowanych i eksportowanych JMS są spójne z opcjami powiązań osadzonych elementów importowanych JMS i MQ JMS produktu WebSphere. Kluczowe opcje obejmują definicje nagłówek i dostęp do istniejących zasobów środowiska Java EE. Jednak z uwagi na ogólną naturę nie istnieją opcje połączeń specyficzne dla określonego dostawcy JMS, a ponadto możliwości generowania przez te powiązania zasobów podczas wdrażania i instalowania są ograniczone.

Ogólne elementy importowane

Podobnie jak w przypadku aplikacji elementu importowanego MQ JMS ogólna implementacja usługi JMS jest asynchroniczna i obsługuje trzy sposoby wywołania: jednokierunkowe, dwukierunkowe (nazywane także żądanie-odpowiedź) oraz wywołanie zwrotne.

Podczas wdrażania elementu importowanego JMS wdrożony zostaje także sterowany komunikatami komponent bean (MDB) udostępniany przez środowisko wykonawcze. Komponent MDB nasłuchuje odpowiedzi na komunikat żądania. Komponent MDB jest powiązany z miejscem docelowym (nasłuchuje go) wysłanym z żądaniem w polu nagłówka replyTo komunikatu JMS.

Ogólne elementy eksportowane

Powiązania ogólnych elementów eksportowanych JMS różnią się od powiązań elementów eksportowanych systemów EIS sposobem zwracania wyniku. Ogólny element eksportowany JMS jawnie wysyła odpowiedź do miejsca docelowego replyTo określonego w komunikacie przychodzącym. Gdy nie jest ono określone, używane jest miejsce docelowe wysyłania.

Podczas wdrażania ogólnego elementu eksportowanego JMS wdrożony zostaje także sterowany komunikatami komponent bean (różny od komponentu MDB używanego na potrzeby ogólnego elementu importowanego JMS). Nasłuchuje on żądań przychodzących w miejscu docelowym odbioru, a następnie rozsyła żądania do przetworzenia przez środowisko wykonawcze SCA.

Nagłówki specjalne

W ogólnych elementach importowanych i eksportowanych JMS stosowane są specjalne właściwości nagłówka zawierające informacje pozwalające powiązaniu docelowemu na właściwą obsługę komunikatu.

Na przykład właściwość TargetFunctionName jest używana przez domyślny selektor funkcji do identyfikowania nazwy operacji w wywoływanym interfejsie elementu eksportowanego.

Uwaga: Powiązanie elementu importowanego można skonfigurować w taki sposób, aby ustawiało nagłówek TargetFunctionName jako nazwę dla każdej operacji.

Zasoby środowiska Java EE

Podczas wdrażania powiązania usługi JMS w środowisku Java EE utworzona zostaje pewna liczba zasobów środowiska Java EE.

- Port procesu nasłuchiwanego na potrzeby nasłuchiwanego miejsca docelowego odbioru (odpowiedź; tylko wywołania dwukierunkowe) w przypadku elementów importowanych oraz miejsca docelowego odbioru (żądanie) w przypadku elementów eksportowanych
- Fabryka połączeń ogólnych JMS na potrzeby połączeń wychodzących (element importowany) oraz połączeń przychodzących (element eksportowany)
- Ogólne miejsce docelowe JMS na potrzeby miejsc docelowych wysyłania (element importowany) i odbioru (element eksportowany; tylko wywołania dwukierunkowe)
- Fabryka połączeń ogólnych JMS na potrzeby połączeń odpowiedzi (tylko wywołania dwukierunkowe i opcjonalne; w przeciwnym razie w przypadku elementów importowanych używane jest połączenie wychodzące, a w przypadku elementów eksportowanych używane jest połączenie przychodzące)
- Ogólne miejsce docelowe JMS na potrzeby miejsc docelowych odbioru (element importowany) i wysyłania (element eksportowany) - tylko wywołania dwukierunkowe
- Domyślne miejsce docelowe wywołań zwrotnych JMS dostawcy przesyłania komunikatów (tylko wywołania dwukierunkowe)
- Domyślna fabryka połączeń wywołań zwrotnych JMS dostawcy przesyłania komunikatów używana w celu uzyskania dostępu do miejsca docelowego wywołań zwrotnych JMS (tylko wywołania dwukierunkowe)
- Miejsce docelowe kolejki wywołań zwrotnych magistrali SIB używane do zapisywania informacji o komunikacie żądania podczas przetwarzania odpowiedzi (tylko wywołania dwukierunkowe)

Czynność instalacji powoduje utworzenie zasobu fabryki połączeń, trzech miejsc docelowych oraz specyfikacji aktywowania na podstawie informacji zawartych w plikach importu i eksportu.

Ogólne nagłówki JMS:

Ogólne nagłówki JMS są obiektami SDO, które zawierają wszystkie właściwości dla ogólnych właściwości komunikatu JMS. Te właściwości mogą pochodzić z komunikatu przychodzącego lub mogą być właściwościami, które zostaną zastosowane do komunikatu wychodzącego.

Komunikat JMS zawiera dwa typy nagłówków: nagłówek systemowy JMS i wiele właściwości JMS. Dostęp do obu typów nagłówków można uzyskać w module mediacji, w obiekcie SMO lub przy użyciu funkcji API ContextService.

Następujące właściwości są ustawione statycznie w klasie methodBinding:

- JMSType,
- JMSCorrelationID,
- JMSDeliveryMode,
- JMSPriority.

Ogólne powiązanie JMS obsługuje również dynamiczne modyfikowanie nagłówków JMS i właściwości w ten sam sposób, co powiązania JMS i MQ JMS.

Niektórzy ogólni dostawcy JMS nakładają ograniczenia właściwości, które mogą być ustawiane przez aplikację oraz określonych kombinacji tych właściwości. Więcej informacji na ten temat należy uzyskać z dokumentacji produktu innej firmy. Do klasy methodBinding została jednak dodana właściwość ignoreInvalidOutboundJMSProperties, która umożliwia propagowanie dowolnych wyjątków.

Ogólne nagłówki JMS i właściwości komunikatu są używane tylko, jeśli włączony jest podstawowy przełącznik powiązania SCDL architektury SCA. Jeśli ten przełącznik jest włączony, propagowane są informacje o kontekście. Domyślnie ten przełącznik jest włączony. Aby uniemożliwić propagowanie informacji o kontekście, należy zmienić jego wartość na **false**.

Jeśli propagowanie kontekstu jest włączone, informacje w nagłówku mogą przepływać do komunikatu lub do komponentu docelowego. Aby włączyć lub wyłączyć propagowanie kontekstu, należy ustawić wartość **true** lub **false** atrybutu contextPropagationEnabled powiązania importu i eksportu. Na przykład:

```
<esbBinding xsi:type="eis:JMSImportBinding" contextProgagationEnabled="true">
```

Wartością domyślną jest **true**.

Rozwiązywanie problemów z ogólnymi powiązaniem JMS:

Istnieje możliwość diagnozowania i rozwiązywania problemów związanych z ogólnym powiązaniem JMS.

Wyjątki implementacji

W odpowiedzi na różne warunki błędu implementacja ogólnego elementu importowanego lub eksportowanego JMS może zwracać wyjątki dwóch typów:

- Wyjątek biznesowy usługi: ten wyjątek jest zwracany po wystąpieniu uszkodzenia określonego w interfejsie biznesowym usługi (typ portu WSDL).
- Wyjątek środowiska wykonawczego usługi: jest zgłaszany we wszystkich innych przypadkach. Najczęściej wyjątek cause (przyczyna) będzie zawierać pierwotny wyjątek (JMSException).

Rozwiązywanie problemów z utratą ważności ogólnego komunikatu JMS

Komunikat żądania wysłany przez dostawcę JMS może utracić ważność.

Utrata ważności żądania odnosi się do utraty ważności komunikatu żądania dostawcy JMS po upływie czasu JMSExpiration określonego dla tego komunikatu żądania. Tak jak w przypadku innych powiązań JMS ogólne powiązanie JMS obsługuje utratę ważności żądania przez ustawianie czasu utraty ważności w komunikacie wywołania zwrotnego generowanym przez element importowany na taką samą wartość, jaką ma żądanie wychodzące. Powiadomienie o utracie ważności komunikatu wywołania zwrotnego wskazuje, że komunikat żądania utracił ważność i należy powiadomić klienta za pomocą wyjątku biznesowego.

Jeśli jednak miejscem docelowym wywołania zwrotnego jest dostawca innej firmy, ten typ określania utraty ważności żądania nie jest obsługiwany.

Utrata ważności odpowiedzi odnosi się do utraty ważności komunikatu odpowiedzi dostawcy JMS po upływie czasu `JMSExpiration` określonego dla tego komunikatu odpowiedzi.

Określanie czasu utraty ważności odpowiedzi związanej z ogólnym powiązaniem JMS nie jest obsługiwane, ponieważ dokładna realizacja utraty ważności dostawcy JMS innej firmy nie jest zdefiniowana. Istnieje jednak możliwość sprawdzenia, czy odpowiedź nie utraciła ważności, po jej ewentualnym odebraniu.

W przypadku wychodzących komunikatów żądania wartość `JMSExpiration` jest obliczana w oparciu o czas oczekiwania oraz wartości `requestExpiration` ustawione w nagłówku `asynchHeader`.

Rozwiązywanie problemów z ogólną fabryką połączeń JMS

Jeśli zdefiniowano pewne typy fabryk połączeń w ogólnym dostawcy JMS, podczas próby uruchomienia aplikacji można odebrać komunikat o błędzie. W celu uniknięcia problemu można zmodyfikować zewnętrzną fabrykę połączeń.

Podczas uruchamiania aplikacji może zostać wyświetlony następujący komunikat o błędzie:

```
MDB Listener Port JMSConnectionFactory type does not match JMSDestination type  
(Typ fabryki połączeń JMS portu nasłuchiwanie MDB jest niezgodny z typem miejsca docelowego JMS)
```

Ten problem może wystąpić wtedy, gdy definiowane są zewnętrzne fabryki połączeń. W szczególności może zostać zgłoszony wyjątek, gdy użytkownik tworzy fabrykę połączeń tematów JMS 1.0.2 zamiast ujednoczonej fabryki połączeń JMS 1.1 (czyli takiej, w której możliwa jest obsługa komunikacji zarówno w trybie punkt z punktem, jak i w trybie publikowania/subskrybowania).

Aby rozwiązać ten problem, wykonaj następujące kroki:

1. Uzyskaj dostęp do używanego ogólnego dostawcy JMS.
2. Zastąp zdefiniowaną fabrykę połączeń tematów JMS 1.0.2 ujednoczoną fabryką połączeń JMS 1.1.

Po uruchomieniu aplikacji z nowo zdefiniowaną fabryką połączeń JMS 1.1 komunikat o błędzie nie powinien być więcej wyświetlany.

Ogólne komunikaty SCA oparte na usługach JMS nie są wyświetlane w menedżerze zdarzeń zakończonych niepowodzeniem

Jeśli komunikaty SCA są wynikiem niepowodzenia ogólnej interakcji JMS, powinny znaleźć się w menedżerze zdarzeń zakończonych niepowodzeniem. Jeśli takie komunikaty nie pojawiają się w menedżerze zdarzeń zakończonych niepowodzeniem, należy upewnić się, że wartość właściwości określającej maksymalną liczbę ponowionych prób bazowego portu nasłuchiwanie jest równa lub większa niż 1. Ustawienie wartości 1 lub większej umożliwia interakcję z menedżerem zdarzeń zakończonych niepowodzeniem w trakcie wywołań SCA związanych z ogólnymi powiązaniem JMS.

Obsługa wyjątków:

Sposób skonfigurowania powiązania określa sposób obsługiwania wyjątków wywołanych przez procedury obsługi danych lub powiązania danych. Na działanie systemu podczas zgłoszenia takiego wyjątku dodatkowy wpływ ma rodzaj przepływu mediacji.

Jeśli procedura obsługi danych lub powiązanie danych zostanie wywołane przez powiązanie użytkownika, może wystąpić wiele różnych problemów. Procedura obsługi danych może na przykład odebrać komunikat z uszkodzonym ładunkiem lub może próbować odczytać komunikat w niepoprawnym formacie.

Sposób obsługi takiego wyjątku przez powiązanie użytkownika jest określany na podstawie tego, w jaki sposób zaimplementowano procedurę obsługi danych lub powiązanie danych. Zalecanym sposobem utworzenia powiązania danych jest zaprojektowanie go tak, aby zgłaszało wyjątek **DataBindingException**.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, każdy wyjątek procedury obsługi danych jest opakowany w wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

W przypadku wywołania wyjątku czasu wykonywania, między innymi wyjątku **DataBindingException**, mają miejsce następujące działania:

- Jeśli przepływ mediacji jest skonfigurowany jako transakcyjny, komunikat JMS jest domyślnie zapisywany w menedżerze zdarzeń zakończonych niepowodzeniem w celu ręcznego powtórzenia lub usunięcia.

Uwaga: Użytkownik może zmienić tryb odtwarzania dla powiązania tak, aby komunikat był wycofywany zamiast zapisywania go w menedżerze zdarzeń zakończonych niepowodzeniem.

- Jeśli przepływ mediacji nie jest transakcyjny, wyjątek jest rejestrowany, a komunikat zostaje utracony.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, wyjątek procedury obsługi danych jest tworzony wewnątrz wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

Powiązania WebSphere MQ JMS

Powiązanie WebSphere MQ JMS udostępnia integrację z zewnętrznymi aplikacjami, które używają dostawcy opartego na specyfikacji WebSphere MQ JMS.

Powiązania eksportu i importu WebSphere MQ JMS służą do bezpośredniej integracji z zewnętrznymi systemami JMS lub MQ JMS z poziomu środowiska serwera użytkownika. To eliminuje konieczność używania opcji łącza MQ lub łącza klienta magistrali integracji usług.

Jeśli komponent współdziała z usługą opartą na specyfikacji WebSphere MQ JMS za pośrednictwem importu, powiązanie importu WebSphere MQ wykorzystuje miejsce docelowe, do którego będą wysyłane dane oraz miejsce docelowe, w którym może odebrać odpowiedź. Konwersja danych na komunikat JMS i z komunikatu JMS jest realizowana za pośrednictwem komponentu krańcowego procedury obsługi danych JMS lub powiązania danych JMS.

Jeśli moduł SCA udostępnia usługę klientom WebSphere MQ JMS, powiązanie eksportu WebSphere MQ JMS wykorzystuje miejsce docelowe, w którym może odebrać żądanie oraz z którego może być wysłana odpowiedź. Konwersja danych na komunikat JMS i z komunikatu JMS jest realizowana za pośrednictwem procedury obsługi danych JMS lub powiązania danych JMS.

Selektor funkcji udostępnia odwzorowanie operacji w komponencie docelowym, który ma zostać wywołany.

Przegląd powiązań WebSphere MQ JMS:

Powiązanie WebSphere MQ JMS udostępnia integrację z aplikacjami zewnętrznymi używającymi dostawcy WebSphere MQ JMS.

Czynności administracyjne dotyczące produktu WebSphere MQ

Administrator systemu WebSphere MQ powinien utworzyć bazowy menedżer kolejek WebSphere MQ, który będzie używany przez powiązania WebSphere MQ JMS, zanim zostanie uruchomiona aplikacja zawierająca te powiązania.

Powiązania importu WebSphere MQ JMS

Import WebSphere MQ JMS umożliwia komponentom w module SCA komunikację z usługami udostępnianymi przez dostawców opartych na specyfikacji WebSphere MQ JMS. Użytkownik musi wykorzystywać obsługiwaną wersję produktu WebSphere MQ. Szczegółowe wymagania sprzętowe i programowe zawierają strony działu wsparcia IBM.

Obsługiwane są dwa typy scenariuszy używania dla powiązań importu WebSphere MQ JMS, zależne od typu wywoływanej operacji:

- **Jednokierunkowy:** import WebSphere MQ JMS umieszcza komunikat w miejscu docelowym wysyłania skonfigurowanym w powiązaniu importu. Do pola replyTo nagłówka JMS nie są wysyłane żadne informacje.
- **Dwukierunkowy (żądanie-odpowieź):** import WebSphere MQ JMS umieszcza komunikat w miejscu docelowym wysyłania.

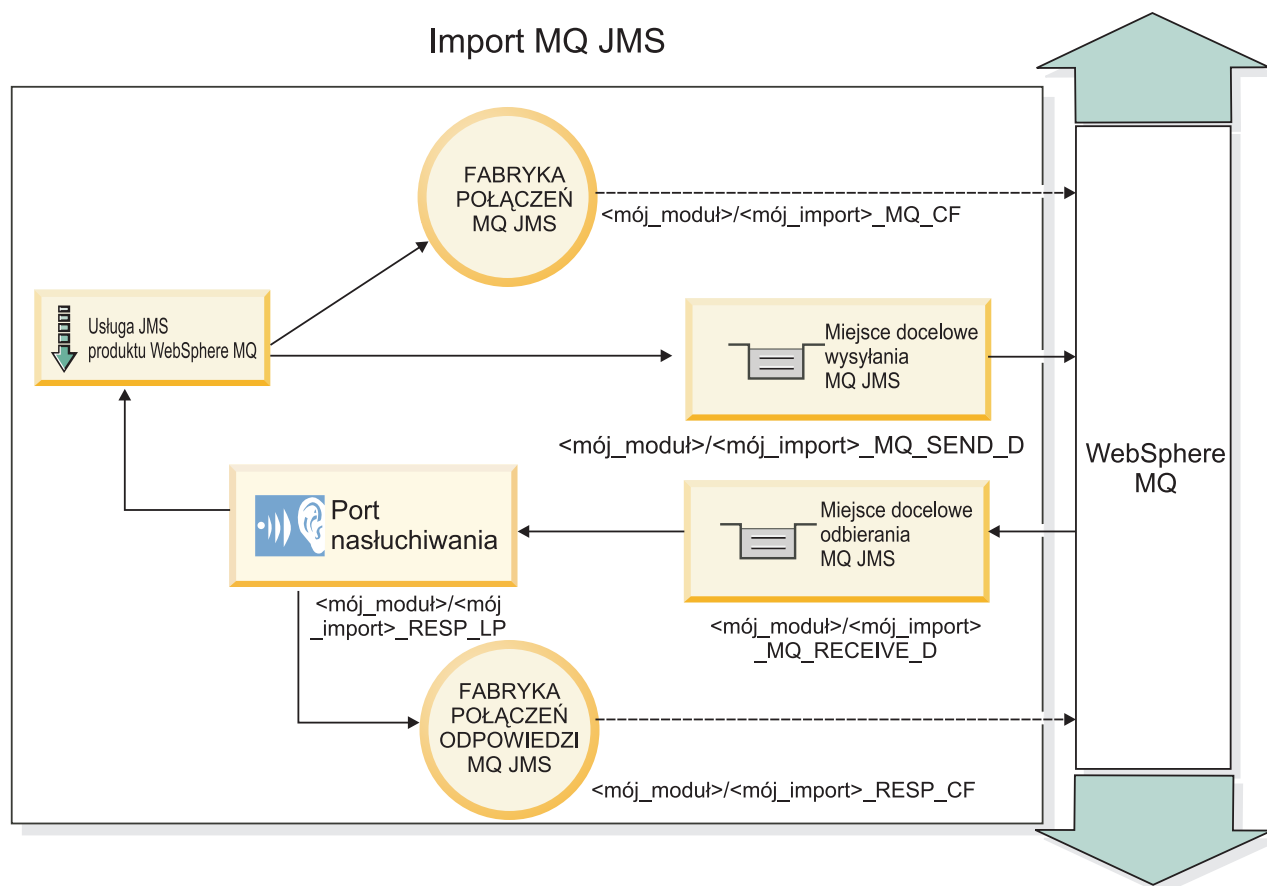
Miejsce docelowe receive (odbioru) jest ustawiane w polu nagłówka replyTo. Wdrażany jest komponent bean sterowany komunikatami (Message Driven Bean - MDB), który nasłuchuje w miejscu docelowym odbioru, a po otrzymaniu odpowiedzi ten komponent MDB przekazuje odpowiedź z powrotem do komponentu.

Powiązanie importu może być skonfigurowane (przy użyciu pola **Schemat korelacji odpowiedzi** w produkcie Integration Designer) tak, aby oczekiwać skopiowania identyfikatora korelacji komunikatu odpowiedzi z identyfikatora komunikatu żądania (jest to ustawienie domyślne) lub z identyfikatora korelacji komunikatu żądania.

Istnieje możliwość określenia dynamicznych i statycznych właściwości nagłówka zarówno dla jednokierunkowych, jak i dwukierunkowych scenariuszy używania. Właściwości statyczne można ustawić z poziomu powiązania metody importu JMS. Niektóre z tych właściwości mają specjalne znaczenie dla środowiska wykonawczego JMS architektury SCA.

Należy pamiętać, że powiązanie WebSphere MQ JMS jest powiązaniem asynchronicznym. Jeśli komponent wywołujący wywołuje import WebSphere MQ JMS w sposób synchroniczny (dla operacji dwukierunkowej), komponent wywołujący jest blokowany do chwili zwrócenia odpowiedzi przez usługę JMS.

Na ilustracji Rys. 37 na stronie 133 przedstawiono sposób połączenia importu z usługą zewnętrzną.



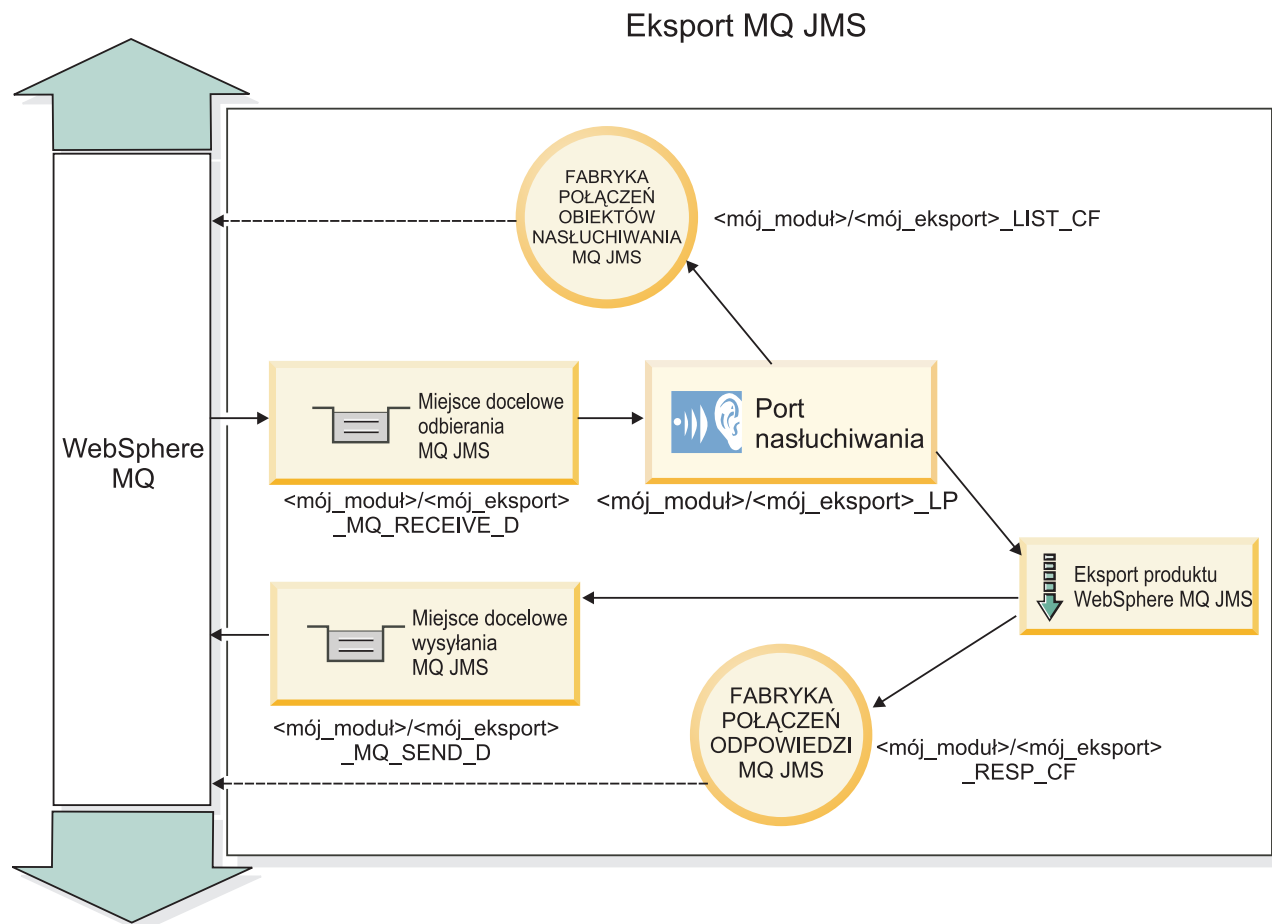
Rysunek 82. Zasoby powiązania importu WebSphere MQ JMS

Powiązania eksportu WebSphere MQ JMS

Powiązanie eksportu WebSphere MQ JMS zapewnia modułom SCA sposób udostępniania usług zewnętrznym aplikacjom JMS w dostawcy JMS opartym na produkcie WebSphere MQ.

Wdrażany jest komponent MDB, który nasłuchuje żądań przychodzących do miejsca docelowego receive (odbioru) określonego w powiązaniu eksportu. Miejsce docelowe określone w polu send służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołany komponent udostępni odpowiedź. Miejsce docelowe określone w polu replyTo komunikatu odpowiedzi nadpisuje miejsce docelowe określone w polu send.

Na ilustracji Rys. 38 na stronie 134 przedstawiono sposób połączenia zewnętrznego requestera z eksportem.



Rysunek 83. Zasoby powiązania eksportu WebSphere MQ JMS

Uwaga: Ilustracje Rys. 37 na stronie 133 i Rys. 38 na stronie 134 przedstawiają sposób połączenia aplikacji z wcześniejszej wersji produktu IBM Business Process Manager z usługą zewnętrzną. W przypadku aplikacji utworzonych dla produktu IBM Business Process Manager 7.0 zamiast portu nasłuchiwanie i fabryki połączeń używana jest specyfikacja aktywowania.

Kluczowe opcje powiązań usług JMS z produktem WebSphere MQ:

Kluczowe opcje powiązań usług JMS z produktem WebSphere MQ obejmują nagłówki, artefakty środowiska Java EE oraz utworzone zasoby środowiska Java EE.

Nagłówki

W nagłówku komunikatu JMS znajduje się pewna liczba predefiniowanych pól zawierających wartości używane zarówno przez klienty, jak i przez dostawców do identyfikowania i kierowania komunikatów. Nagłówki można konfigurować ustalonymi wartościami za pomocą właściwości powiązań lub można je określać dynamicznie w czasie wykonywania.

JMSCorrelationID

Odsyła do komunikatu pokrewnego. Wartością tego pola jest zwykle łańcuch identyfikatora komunikatu, na który dany komunikat jest odpowiedzią.

TargetFunctionName

Ten nagłówek jest używany przez jeden z dostępnych selektorów funkcji do identyfikowania wywoływanej operacji. Ustawienie właściwości nagłówka TargetFunctionName usługi JMS w komunikacie wysłanym do elementu eksportowanego JMS pozwala użyć tego selektora funkcji. Tę właściwość można ustawiać bezpośrednio w aplikacjach klienckich JMS lub podczas nawiązywania połączenia z elementem importowanym z powiązaniem usługi JMS do takiego elementu eksportowanego. W takim przypadku należy skonfigurować powiązanie elementu importowanego JMS w taki sposób, aby ustawiało nagłówek TargetFunctionName dla każdej operacji w interfejsie na nazwę operacji.

Schematy korelacji

Powiązania usługi JMS z produktem WebSphere MQ udostępniają różne schematy korelacji, które są używane do określania, w jaki sposób należy korelować komunikaty żądania z komunikatami odpowiedzi.

RequestMsgIDToCorreIID

Do pola JMSCorrelationID kopiowana jest wartość pola JMSMessageID. To jest ustawienie domyślne.

RequestCorreIIDToCorreIID

Do pola JMSCorrelationID kopiowana jest wartość pola JMSCorrelationID.

Zasoby środowiska Java EE

Podczas wdrażania elementu importowanego MQ JMS w środowisku Java EE utworzona zostaje pewna liczba zasobów środowiska Java EE.

Parametry

Fabryka połączeń MQ

Używana przez klienty do tworzenia połączenia z dostawcą MQ JMS.

Fabryka połączeń odpowiedzi

Używana przez środowisko wykonawcze MQ JMS architektury SCA, gdy miejsce docelowe wysyłania znajduje się w innym menedżerze kolejek niż miejsce docelowe odbioru.

Specyfikacja aktywowania

Specyfikacja aktywowania MQ JMS jest powiązana z jednym lub kilkoma komponentami bean sterowanymi komunikatami i udostępnia konfigurację pozwalającą im odbierać komunikaty.

Miejsca docelowe

- Miejsce docelowe wysyłania:
 - Elementy importowane: miejsce wysyłania żądania lub komunikatu wychodzącego.
 - Elementy eksportowane: miejsce, do którego jest wysyłany komunikat odpowiedzi, jeśli nie jest zastąpione za pomocą pola nagłówka JMSReplyTo w komunikacie przychodzącym.
- Miejsce docelowe odbioru:
 - Elementy importowane: miejsce, w którym należy umieścić odpowiedź lub komunikat przychodzący.
 - Elementy eksportowane: miejsce, w którym należy umieścić komunikat przychodzący lub żądanie.

Nagłówki JMS:

Komunikat JMS zawiera dwa typy nagłówków: nagłówek systemowy JMS i wiele właściwości JMS. Dostęp do obu typów nagłówków można uzyskać w module mediacji, w obiekcie SMO lub przy użyciu funkcji API ContextService.

Nagłówek systemu JMS

Nagłówek systemu JMS jest reprezentowany w obiekcie SMO przez element JMSHeader, który zawiera wszystkie pola znajdujące się zazwyczaj w nagłówku JMS. Mimo że mogą one być modyfikowane podczas mediacji (lub w funkcji ContextService), niektóre pola nagłówka systemu JMS ustawione w obiekcie SMO nie będą propagowane w wychodzącym komunikacie JMS, ponieważ są nadpisywane przez wartości systemowe lub statyczne.

Kluczowymi polami w nagłówku systemu JMS, które mogą być aktualizowane podczas mediacji (lub w funkcji ContextService), są:

- **JMSType** i **JMSCorrelationID** – wartości konkretnych właściwości predefiniowanego nagłówka komunikatu.
- **JMSDeliveryMode** – wartości dla trybu dostarczania (persistent - trwałe lub nonpersistent - nietrwałe; wartość domyślna to persistent).
- **JMSPriority** – wartość priorytetu (od 0 do 9, wartość domyślna to JMS_Default_Priority).

Właściwości JMS

Właściwości JMS są reprezentowane w obiekcie SMO jako pozycje na liście Properties (właściwości). Właściwości można dodawać, aktualizować lub usuwać podczas mediacji lub przy użyciu funkcji API ContextService.

Właściwości mogą być też ustawiane statycznie w powiązaniu JMS. Właściwości ustawione statycznie nadpisują ustawienia (o tej samej nazwie), które są ustawione dynamicznie.

Właściwości użytkownika propagowane z innych powiązań (na przykład powiązania HTTP) będą wyprowadzane w powiązaniu JMS jako właściwości JMS.

Ustawienia propagowania nagłówka

Propagowanie nagłówka systemu JMS i właściwości z przychodzącego komunikatu JMS do kolejnych komponentów lub z wcześniejszych komponentów do wychodzącego komunikatu JMS może być kontrolowane za pośrednictwem flagi Propaguj nagłówek protokołu w powiązaniu.

Jeśli flaga Propaguj nagłówek protokołu jest ustawiona, informacje w nagłówku mogą przepływać do komunikatu lub do komponentu docelowego, jak opisano poniżej:

- **Żądanie eksportu JMS**
Nagłówek JMS odebrany w komunikacie będzie propagowany do komponentów docelowych za pośrednictwem usługi kontekstu. Właściwości JMS odebrane w komunikacie będą propagowane do komponentów docelowych za pośrednictwem usługi kontekstu.
- **Odpowiedź eksportu JMS**
Wszystkie pola nagłówka JMS ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu eksportu JMS. Wszystkie właściwości ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu eksportu JMS.
- **Żądanie importu JMS**
Wszystkie pola nagłówka JMS ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu importu JMS. Wszystkie właściwości ustawione w usłudze kontekstu będą używane w komunikacie wychodzącym, jeśli nie zostaną nadpisane przez właściwości statyczne ustawione w powiązaniu importu JMS.
- **Odpowiedź importu JMS**

Nagłówek JMS odebrany w komunikacie będzie propagowany do komponentów docelowych za pośrednictwem usługi kontekstu. Właściwości JMS odebrane w komunikacie będą propagowane do komponentów docelowych za pośrednictwem usługi kontekstu.

Klienci zewnętrzne:

Serwer może wysyłać komunikaty do klientów zewnętrznych i odbierać komunikaty od nich za pośrednictwem powiązań WebSphere MQ JMS.

Klient zewnętrzny (na przykład portal WWW lub system EIS) może wysyłać komunikat do komponentu SCA w aplikacji za pośrednictwem eksportu lub może być wywoływany przez komponent SCA w aplikacji za pośrednictwem importu.

Powiązanie eksportu WebSphere MQ JMS wdraża komponenty bean sterowane komunikatami (Message Driven Bean - MDB) w celu nasłuchiwania żądań przychodzących do miejsca docelowego odbioru określonego w powiązaniu eksportu. Miejsce docelowe określone w polu send służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołana aplikacja udostępni odpowiedź. Dzięki temu klient zewnętrzny może wywoływać aplikacje za pośrednictwem powiązania eksportu.

Importy WebSphere MQ JMS są związane z klientami zewnętrznymi i mogą dostarczać do nich komunikaty. Komunikat może wymagać odpowiedzi od klienta zewnętrznego, ale może też nie wymagać tej odpowiedzi.

Więcej informacji na temat interakcji z klientami zewnętrznymi przy użyciu produktu WebSphere MQ można znaleźć w Centrum informacyjnym produktu WebSphere MQ.

Rozwiązywanie problemów z powiązaniem JMS produktu WebSphere MQ:

Istnieje możliwość diagnozowania i rozwiązywania problemów dotyczących powiązań JMS produktu WebSphere MQ.

Wyjątki implementacji

W odpowiedzi na różne warunki błędów implementacja elementu importowanego lub eksportowanego MQ JMS może zwracać wyjątki dwóch typów:

- Wyjątek biznesowy usługi: ten wyjątek jest zwracany po wystąpieniu uszkodzenia określonego w interfejsie biznesowym usługi (typ portu WSDL).
- Wyjątek środowiska wykonawczego usługi: jest zgłaszany we wszystkich innych przypadkach. Najczęściej wyjątek cause (przyczyna) będzie zawierał pierwotny wyjątek (JMSEException).

Element importowany może na przykład oczekiwać tylko jednego komunikatu odpowiedzi na każdy komunikat żądania. Jeśli nadejdzie więcej niż jedna odpowiedź lub nadejdzie odpowiedź spóźniona (czas utraty ważności odpowiedzi SCA minął), zgłoszony zostanie wyjątek środowiska wykonawczego usługi. Transakcja zostanie wycofana, a komunikat odpowiedzi zostanie usunięty z kolejki lub obsłużony przez menedżer zdarzeń zakończonych niepowodzeniem.

Komunikaty SCA oparte na usługach JMS produktu WebSphere MQ nie są wyświetlane w menedżerze zdarzeń zakończonych niepowodzeniem

Jeśli komunikaty SCA są wynikiem niepowodzenia interakcji usług JMS produktu WebSphere MQ, powinny znaleźć się w menedżerze zdarzeń zakończonych niepowodzeniem. Jeśli takie komunikaty nie pojawiają się w menedżerze zdarzeń zakończonych niepowodzeniem, należy upewnić się, że wartość właściwości określającej maksymalną liczbę ponowionych prób bazowego portu nasłuchiwania jest równa lub większa niż 1. Ustawienie wartości 1 lub większej umożliwi interakcję z menedżerem zdarzeń zakończonych niepowodzeniem w trakcie wywołań SCA dotyczących powiązań MQ JMS.

Scenariusze niepoprawnego użycia: porównanie z powiązaniem produktu WebSphere MQ

Powiązanie JMS produktu WebSphere MQ zostało zaprojektowane z myślą o współpracy z aplikacjami JMS wdrożonymi z produktem WebSphere MQ, który ujawnia komunikaty zgodnie z modelem komunikatów JMS. Elementy importowane i eksportowane produktu WebSphere MQ zostały jednak zaprojektowane głównie na potrzeby współpracy z rodzimymi aplikacjami produktu WebSphere MQ i ujawniają mediacjom pełną treść komunikatu produktu WebSphere MQ.

W następujących scenariuszach należy stosować powiązania JMS produktu WebSphere MQ JMS zamiast powiązań produktu WebSphere MQ:

- Wywoływanie komponentu bean sterowanego komunikatami JMS (MDB) z modułu SCA, gdy komponent MDB jest wdrożony w dostawcy JMS produktu WebSphere MQ. Należy użyć elementu importowanego JMS produktu WebSphere MQ.
- Umożliwienie wywołania modułu SCA z serwletu komponentów Java EE lub z komponentu EJB za pomocą usług JMS. Należy użyć elementu eksportowanego JMS produktu WebSphere MQ.
- Mediowanie treści komunikatu JMS MapMessage w trakcie przejścia przez produkt WebSphere MQ. Należy użyć elementu eksportowanego i elementu importowanego JMS produktu WebSphere MQ w połączeniu z odpowiednią procedurą obsługi danych lub odpowiednim powiązaniem danych.

Są takie sytuacje, w których powiązanie produktu WebSphere MQ i powiązanie JMS produktu WebSphere MQ JMS powinny współpracować. W szczególności podczas łączenia aplikacji Java EE i innych niż Java EE w produkcie WebSphere MQ należy używać elementu eksportowanego produktu WebSphere MQ oraz elementu importowanego JMS produktu WebSphere MQ (lub odwrotnie) w połączeniu z odpowiednimi powiązaniem danych lub modułami mediacji (bądź dwoma tymi rodzajami obiektów).

Obsługa wyjątków:

Sposób skonfigurowania powiązania określa sposób obsługi wyjątków wywołanych przez procedury obsługi danych lub powiązania danych. Na działanie systemu podczas zgłoszenia takiego wyjątku dodatkowy wpływ ma rodzaj przepływu mediacji.

Jeśli procedura obsługi danych lub powiązanie danych zostanie wywołane przez powiązanie użytkownika, może wystąpić wiele różnych problemów. Procedura obsługi danych może na przykład odebrać komunikat z uszkodzonym ładunkiem lub może próbować odczytać komunikat w niepoprawnym formacie.

Sposób obsługi takiego wyjątku przez powiązanie użytkownika jest określany na podstawie tego, w jaki sposób zaimplementowano procedurę obsługi danych lub powiązanie danych. Zalecanym sposobem utworzenia powiązania danych jest zaprojektowanie go tak, aby zgłaszało wyjątek **DataBindingException**.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, każdy wyjątek procedury obsługi danych jest opakowany w wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

W przypadku wywołania wyjątku czasu wykonywania, między innymi wyjątku **DataBindingException**, mają miejsce następujące działania:

- Jeśli przepływ mediacji jest skonfigurowany jako transakcyjny, komunikat JMS jest domyślnie zapisywany w menedżerze zdarzeń zakończonych niepowodzeniem w celu ręcznego powtórzenia lub usunięcia.

Uwaga: Użytkownik może zmienić tryb odtwarzania dla powiązania tak, aby komunikat był wycofywany zamiast zapisywania go w menedżerze zdarzeń zakończonych niepowodzeniem.

- Jeśli przepływ mediacji nie jest transakcyjny, wyjątek jest rejestrowany, a komunikat zostaje utracony.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, wyjątek procedury obsługi danych jest tworzony wewnątrz wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

Powiązania produktu WebSphere MQ

Powiązanie WebSphere MQ udostępnia połączenia architektury SCA (Service Component Architecture) z aplikacjami WebSphere MQ.

Powiązania eksportu i importu WebSphere MQ służą do bezpośredniej integracji z systemem opartym na produkcie WebSphere MQ z poziomu środowiska serwera użytkownika. To eliminuje konieczność używania opcji łącza MQ lub łącza klienta magistrali integracji usług.

Jeśli komponent współdziała z usługą WebSphere MQ za pośrednictwem importu, powiązanie importu WebSphere MQ używa kolejki, do której będą wysyłane dane oraz kolejki, w której może odebrać odpowiedź.

Jeśli moduł SCA udostępnia usługę klientom WebSphere MQ, powiązanie eksportu WebSphere MQ korzysta z kolejki, w której może odebrać żądanie oraz z której może być wysłana odpowiedź. Selektor funkcji udostępnia odwzorowanie operacji w komponencie docelowym, który ma zostać wywołany.

Konwersja danych ładunku z komunikatu MQ i na komunikat MQ jest realizowana za pośrednictwem procedury obsługi danych treści MQ lub powiązania danych treści MQ. Konwersja danych nagłówka na komunikat MQ i z komunikatu MQ jest realizowana za pośrednictwem powiązanie danych nagłówka MQ.

Informacje na temat obsługiwanych wersji produktu WebSphere MQ można znaleźć na stronie WWW zawierającej szczegółowe wymagania systemowe.

Przegląd powiązań WebSphere MQ:

Powiązanie WebSphere MQ udostępnia integrację z rodzimymi aplikacjami opartymi na produkcie MQ.

Czynności administracyjne dotyczące produktu WebSphere MQ

Administrator systemu WebSphere MQ powinien utworzyć bazowy menedżer kolejek WebSphere MQ, który będzie używany przez powiązania WebSphere MQ, zanim zostanie uruchomiona aplikacja zawierająca te powiązania.

Czynności administracyjne dotyczące produktu WebSphere

Należy ustawić właściwość **Rodzima ścieżka do biblioteki** adaptera zasobu MQ w produkcie WebSphere na wartość odpowiadającą wersji produktu WebSphere MQ obsługiwanej przez serwer, a następnie zrestartować serwer. Gwarantuje to, że używane będą biblioteki z obsługiwanej wersji produktu WebSphere MQ. Szczegółowe wymagania sprzętowe i programowe zawierają strony działu wsparcia IBM.

Powiązania importu WebSphere MQ

Powiązanie importu WebSphere MQ umożliwia komponentom w module SCA komunikację z usługami udostępnianymi przez zewnętrzne aplikacje oparte na produkcie WebSphere MQ. Użytkownik musi wykorzystywać obsługiwaną wersję produktu WebSphere MQ. Szczegółowe wymagania sprzętowe i programowe zawierają strony działu wsparcia IBM.

Interakcja z zewnętrznymi systemami WebSphere MQ obejmuje używanie kolejek do wysyłania żądań i odbierania odpowiedzi.

Obsługiwane są dwa typy scenariuszy używania dla powiązań importu WebSphere MQ, zależne od typu wywoływanej operacji:

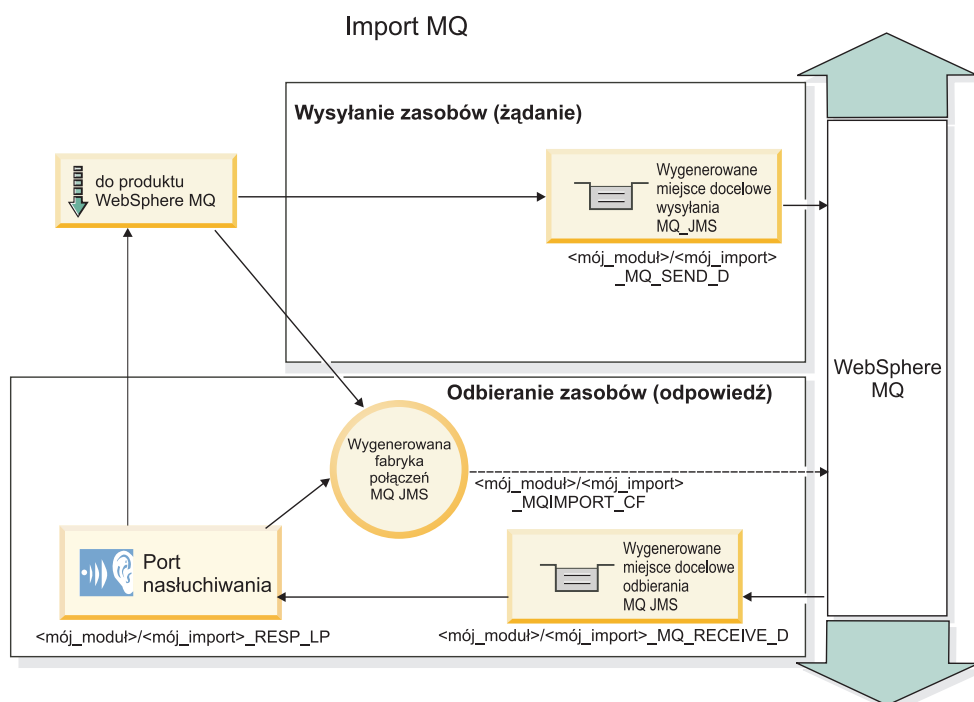
- **Jednokierunkowy:** import WebSphere MQ umieszcza komunikat w kolejce skonfigurowanej w polu **Kolejka miejsca docelowego wysyłania** w powiązaniu importu. Do pola replyTo nagłówka MQMD nie są wysyłane żadne informacje.
- **Dwukierunkowy (żądanie-odpowiedź):** import WebSphere MQ umieszcza komunikat w kolejce skonfigurowanej w polu **Kolejka miejsca docelowego wysyłania**.

Kolejka receive (odbioru) jest ustawiana w polu replyTo nagłówka MQMD. Wdrażany jest komponent bean sterowany komunikatami (Message Driven Bean - MDB), który nasłuchuje w kolejce odbiorczej, a po otrzymaniu odpowiedzi ten komponent MDB przekazuje odpowiedź z powrotem do komponentu.

Powiązanie importu może być skonfigurowane (przy użyciu pola **Schemat korelacji odpowiedzi**) tak, aby oczekiwać skopiowania identyfikatora korelacji komunikatu odpowiedzi z identyfikatora komunikatu żądania (jest to ustawienie domyślne) lub z identyfikatora korelacji komunikatu żądania.

Należy pamiętać, że powiązanie WebSphere MQ jest powiązaniem asynchronicznym. Jeśli komponent wywołujący wywoła import WebSphere MQ w sposób synchroniczny (dla operacji dwukierunkowej), komponent wywołujący jest blokowany do chwili zwrócenia odpowiedzi przez usługę WebSphere MQ.

Na ilustracji Rys. 39 na stronie 140 przedstawiono sposób połączenia importu z usługą zewnętrzną.



Rysunek 84. Zasoby powiązania importu WebSphere MQ

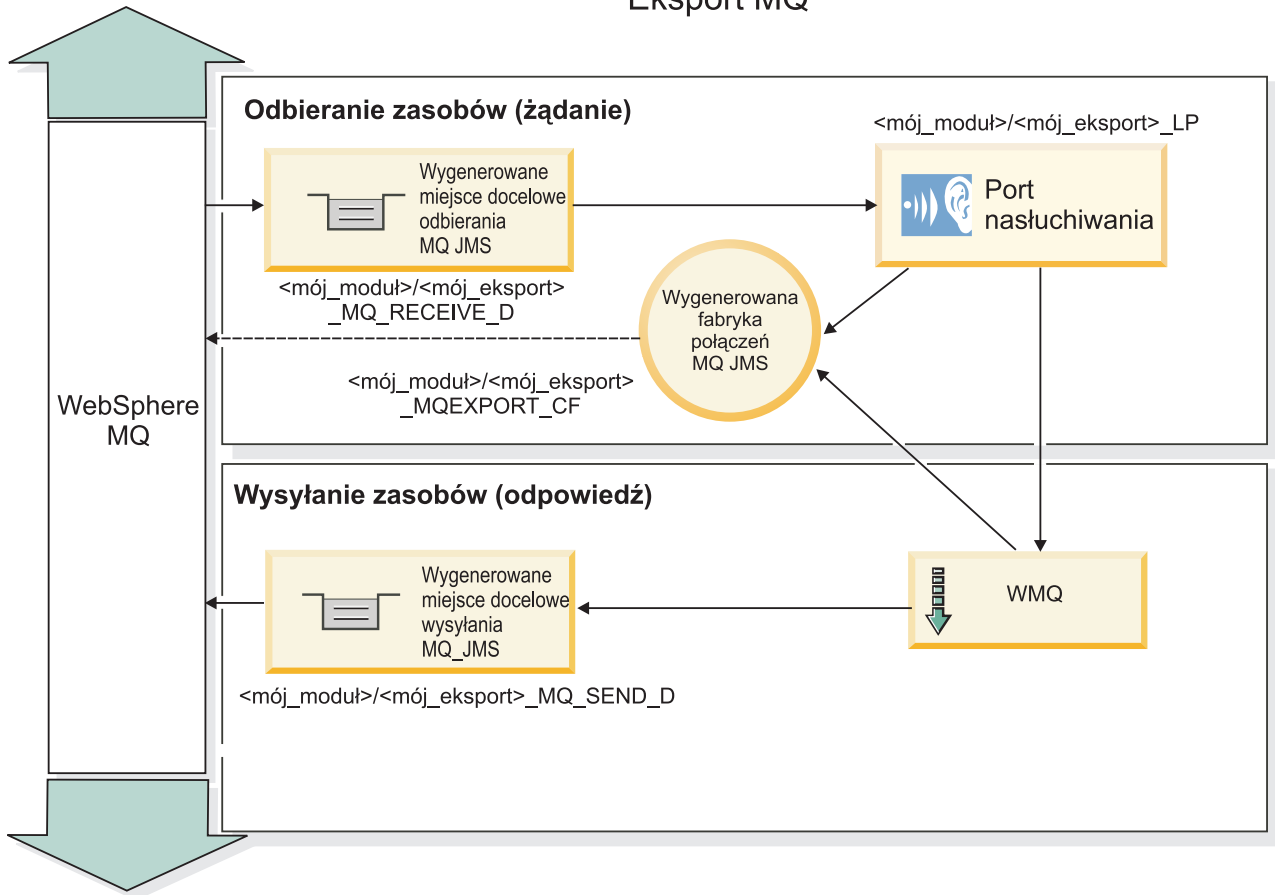
Powiązania eksportu WebSphere MQ

Powiązanie eksportu WebSphere MQ zapewnia modułom SCA sposób udostępniania usług zewnętrznym aplikacjom opartym na produkcie WebSphere MQ.

Wdrażany jest komponent MDB, który nasłuchuje żądań przychodzących do kolejki **Kolejka miejsca docelowego odbioru** określonej w powiązaniu eksportu. Kolejka określona w polu **Kolejka miejsca docelowego wysyłania** służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołujący komponent udostępnia odpowiedź. Kolejka określona w polu replyTo komunikatu odpowiedzi nadpisuje kolejkę określoną w polu **Kolejka miejsca docelowego wysyłania**.

Na ilustracji Rys. 40 na stronie 141 przedstawiono sposób połączenia zewnętrznego requestera z eksportem.

Eksport MQ



Rysunek 85. Zasoby powiązania eksportu WebSphere MQ

Uwaga: Ilustracje Rys. 39 na stronie 140 i Rys. 40 na stronie 141 przedstawiają sposób połączenia aplikacji z wcześniejszej wersji produktu IBM Business Process Manager z usługą zewnętrzną. W przypadku aplikacji utworzonych dla produktu IBM Business Process Manager w wersji 7.x lub nowszej zamiast portu procesu nasłuchiwanie i fabryki połączeń używana jest specyfikacja aktywowania.

Kluczowe opcje powiązania z produktem WebSphere MQ:

Kluczowe opcje powiązania z produktem WebSphere MQ obejmują nagłówki, artefakty środowiska Java EE oraz utworzone zasoby środowiska Java EE.

Schematy korelacji

Aplikacja typu żądanie/odpowiedź produktu WebSphere MQ może używać jednej z wielu technik korelowania komunikatów odpowiedzi z zadaniami korzystających z pól struktury MQMD MessageID i CorrelID. W zdecydowanej większości przypadków requester pozwala menedżerowi kolejek wybrać pole MessageID i oczekuje, że aplikacja odpowiadająca skopiuje jego wartość do pola CorrelID w odpowiedzi. Najczęściej requester i aplikacja odpowiadająca uzyskują niejawnie informacje o aktualnie używanej technice korelacji. Czasami aplikacja odpowiadająca uwzględni różne opcje znajdujące się w polu Report zadaniami opisujące sposób obsługi tych pól.

Powiązania elementów eksportowanych związanych z komunikatami produktu WebSphere MQ można skonfigurować z następującymi opcjami:

Opcje pola MsgId odpowiedzi:

Nowa wartość MsgID

Pozwala menedżerowi kolejek wybrać unikalną wartość pola MsgId w odpowiedzi (opcja domyślna).

Kopiowanie wartości MsgID z żądania

Powoduje skopiowanie wartości pola MsgId z pola MsgId w żądaniu.

Kopiowanie z komunikatu SCA

Powoduje ustawienie wartości pola MsgId na przenoszona w nagłówkach komunikatów odpowiedzi SCA produktu WebSphere MQ lub pozwala menedżerowi kolejek zdefiniować nową wartość Id, gdy ta wartość nie istnieje.

Jak opcje pola Report

Powoduje zbadanie pola Report struktury MQMD w żądaniu w celu wykrycia sposobu obsługi pola MsgId. Obsługiwane są opcje MQRO_NEW_MSG_ID i MQRO_PASS_MSG_ID. Ich działanie jest podobne, jak w przypadku opcji Nowa wartość MsgId i Kopiowanie wartości MsgID z żądania.

Opcje pola CorrelId odpowiedzi:

Kopiowanie wartości MsgID z żądania

Powoduje skopiowanie wartości pola CorrelId z pola MsgId w żądaniu (opcja domyślna).

Kopiowanie wartości CorrelID z żądania

Powoduje skopiowanie wartości pola CorrelId z pola CorrelId w żądaniu.

Kopiowanie z komunikatu SCA

Powoduje ustawienie wartości pola CorrelId na przenoszona w nagłówkach komunikatów odpowiedzi SCA produktu WebSphere MQ lub pozostawienie pustego pola, gdy ta wartość nie istnieje.

Jak opcje pola Report

Powoduje zbadanie pola Report struktury MQMD w żądaniu w celu wykrycia sposobu obsługi pola CorrelId. Obsługiwane są opcje MQRO_COPY_MSG_ID_TO_CORREL_ID i MQRO_PASS_CORREL_ID. Ich działanie jest podobne, jak w przypadku opcji Kopiowanie wartości MsgID z żądania i Kopiowanie wartości CorrelID z żądania.

Powiązania elementów importowanych związanych z komunikatami produktu WebSphere MQ można skonfigurować z następującymi opcjami:

Opcje pola MsgId żądania:

Nowa wartość MsgID

Pozwala menedżerowi kolejek wybrać unikalną wartość pola MsgId w żądaniu (opcja domyślna).

Kopiowanie z komunikatu SCA

Powoduje ustawienie wartości pola MsgId na przenoszona w nagłówkach komunikatów żądania SCA produktu WebSphere MQ lub pozwala menedżerowi kolejek zdefiniować nową wartość Id, gdy ta wartość nie istnieje.

Opcje korelacji odpowiedzi:

Odpowiedź ma wartość CorrelID skopiowaną z pola MsgId

Oczekuje się, że komunikat odpowiedzi będzie mieć ustawioną wartość pola CorrelId na podstawie pola MsgId żądania (opcja domyślna).

Odpowiedź ma wartość MsgID skopiowaną z pola MsgId

Oczekuje się, że komunikat odpowiedzi będzie mieć ustawioną wartość pola MsgId na podstawie pola MsgId żądania.

Odpowiedź ma wartość CorrelID skopiowaną z pola CorrelId

Oczekuje się, że komunikat odpowiedzi będzie mieć ustawioną wartość pola CorrelId na podstawie pola CorrelId żądania.

Zasoby środowiska Java EE

Podczas wdrażania powiązania z produktem WebSphere MQ w środowisku Java EE utworzona zostaje pewna liczba zasobów środowiska Java EE.

Parametry

Fabryka połączeń MQ

Używana przez klienty do tworzenia połączenia z dostawcą WebSphere MQ.

Fabryka połączeń odpowiedzi

Używana przez środowisko wykonawcze MQ architektury SCA, gdy miejsce docelowe wysyłania znajduje się w innym menedżerze kolejek niż miejsce docelowe odbioru.

Specyfikacja aktywowania

Specyfikacja aktywowania MQ JMS jest powiązana z jednym lub kilkoma komponentami bean sterowanymi komunikatami i udostępnia konfigurację pozwalającą im odbierać komunikaty.

Miejsca docelowe

- Miejsce docelowe wysyłania: miejsce, do którego jest wysyłane żądanie lub komunikat wychodzący (element importowany); miejsce, do którego jest wysyłany komunikat odpowiedzi (element eksportowany), jeśli nie jest zastąpione za pomocą pola nagłówka ReplyTo struktury MQMD w komunikacie przychodzącym.
- Miejsce docelowe odbioru: miejsce, w którym należy umieścić odpowiedź/żądanie lub komunikat przychodzący.

Nagłówki WebSphere MQ:

Nagłówki WebSphere MQ uwzględniają określone konwencje podczas konwersji na komunikaty SCA (Service Component Architecture).

Komunikaty WebSphere MQ składają się z nagłówka systemu (struktury MQMD), opcjonalnych innych nagłówków MQ (systemu lub niestandardowych) oraz treści komunikatu. Jeśli w komunikacie istnieje wiele nagłówków komunikatu, kolejność nagłówków ma znaczenie.

Każdy nagłówek zawiera informacje opisujące strukturę kolejnego nagłówka. Struktura MQMD opisuje pierwszy nagłówek.

Sposób analizowania nagłówków MQ

Do analizowania nagłówków MQ używane jest powiązanie danych nagłówka MQ. Następujące nagłówki są obsługiwane automatycznie:

- MQRFH,
- MQRFH2,
- MQCIH,
- MQIIH.

Nagłówki rozpoczynające się od **MQH** są obsługiwane w inny sposób. Konkretnie pola nagłówka nie są analizowane i pozostają one w postaci nieprzeanalizowanych bajtów.

W przypadku innych nagłówków MQ można napisać niestandardowe powiązania danych nagłówka MQ, aby analizować te nagłówki.

Sposób uzyskiwania dostępu do nagłówków MQ

Dostęp do nagłówków MQ w produkcie może być uzyskiwany na jeden z dwóch sposobów:

- za pośrednictwem obiektu SMO w mediacji,

- za pośrednictwem funkcji API ContextService.

Nagłówki MQ są reprezentowane wewnętrznie przy użyciu elementu MQHeader obiektu SMO. Element MQHeader jest kontenerem na dane nagłówka, który rozszerza element MQControl, ale zawiera element wartości anyType. Zawiera on strukturę MQMD, element MQControl (informacje sterujące treści komunikatu MQ) oraz listę innych nagłówków MQ.

- Struktura MQMD reprezentuje treść opisu komunikatu WebSphere MQ, z wyjątkiem informacji określających strukturę i kodowanie treści.
- Element MQControl zawiera informacje określające strukturę i kodowanie treści komunikatu.
- Element MQHeaders zawiera listę obiektów MQHeader.

Łańcuch nagłówka MQ jest rozwinięty, a więc każdy nagłówek MQ w obiekcie SMO przynosi własne informacje sterujące (identyfikator CCSID, kodowanie i format). Nagłówki można w łatwy sposób dodawać lub usuwać, bez zmiany innych danych nagłówków.

Ustawianie pól w strukturze MQMD

Strukturę MQMD można zaktualizować przy użyciu funkcji API Context lub za pośrednictwem obiektu SMO w mediacji. Do wychodzącego komunikatu MQ automatycznie propagowane są następujące pola:

- Encoding,
- CodedCharacterSet,
- Format,
- Report,
- Expiry,
- Feedback,
- Priority,
- Persistence,
- CorrelId,
- MsgFlags.

Aby propagować następujące właściwości do wychodzącego komunikatu MQ, należy skonfigurować powiązanie MQ importu lub eksportu:

MsgID

Opcję **Identyfikator komunikatu żądania** należy ustawić na wartość kopii z komunikatu SCA.

MsgType

Należy anulować zaznaczenie pola wyboru **Ustaw typ komunikatu na MQMT_DATAGRAM lub MQMT_REQUEST w przypadku operacji żądanie-odpowiedź**.

ReplyToQ

Należy anulować zaznaczenie pola wyboru **Nadpisz kolejkę odpowiedzi komunikatu żądania**.

ReplyToQMgr

Należy anulować zaznaczenie pola wyboru **Nadpisz kolejkę odpowiedzi komunikatu żądania**.

Od wersji 7.0 produktu pola kontekstu można nadpisywać przy użyciu właściwości definiowanej w definicji miejsca docelowego JNDI. Aby propagować następujące pola do wychodzącego komunikatu MQ, należy ustawić właściwość definiowaną MDCTX na wartość SET_IDENTITY_CONTEXT dla miejsca docelowego wysyłania:

- UserIdentifier,
- AppIdentityData.

Aby propagować następujące właściwości do wychodzącego komunikatu MQ, należy ustawić właściwość definiowaną MDCTX na wartość SET_ALL_CONTEXT dla miejsca docelowego wysyłania:

- UserIdentifier,
- ApplIdentityData,
- PutApplType,
- PutApplName,
- ApplOriginData.

Niektóre pola nie są propagowane do wychodzącego komunikatu MQ. Następujące pola są nadpisywane podczas wysyłania komunikatu:

- BackoutCount,
- AccountingToken,
- PutDate,
- PutTime,
- Przesunięcie
- OriginalLength.

Statyczne dodawanie nagłówka MQCIH w wiązaniu WebSphere MQ:

Program IBM Business Process Manager obsługuje statyczne dodawanie informacji nagłówka MQCIH bez stosowania modułu mediacji.

Są różne sposoby dodawania informacji nagłówka MQCIH do komunikatu (na przykład wykorzystanie operacji podstawowej mediacji Header Setter). Czasami może być użyteczne dodanie informacji nagłówka statycznie, bez wykorzystywania dodatkowego modułu mediacji. Statyczny nagłówek zawierający nazwę programu CICS, identyfikator transakcji i inne szczegóły nagłówka formatu danych, można zdefiniować i dodać jako część powiązania WebSphere MQ.

Aby można było statycznie dodać informacje nagłówka MQCIH, programy WebSphere MQ, MQ CICS Bridge i CICS muszą być odpowiednio skonfigurowane.

Do skonfigurowania importu WebSphere MQ z wartościami statycznymi, które są wymagane dla informacji nagłówka MQCIH, można użyć programu Integration Designer.

Gdy komunikat dotrze i jest przetwarzany przez import WebSphere MQ, następuje sprawdzenie, czy informacje nagłówka MQCIH znajdują się już w komunikacie. Jeśli nagłówek MQCIH już jest, używane są wartości statyczne zdefiniowane w imporcie WebSphere MQ, które zastępują odpowiadające im wartości dynamiczne z komunikatu. Jeśli nagłówka MQCIH nie ma, jest on tworzony w komunikacie i dodawane są do niego statyczne wartości zdefiniowane w imporcie WebSphere MQ.

Statyczne wartości zdefiniowane w imporcie WebSphere MQ odnoszą się do konkretnej metody. Dla różnych metod w tym samym imporcie WebSphere MQ można podać różne statyczne wartości nagłówka MQCIH.

To ułatwienie nie jest używane do udostępniania wartości domyślnych, jeśli nagłówek MQCIH nie zawiera konkretnych informacji nagłówka, ponieważ wartość statyczna zdefiniowana w imporcie WebSphere MQ zastępuje odpowiadającą jej wartość w komunikacie przychodzącym.

Klienci zewnętrzne:

Produkt IBM Business Process Manager może wysyłać komunikaty do klientów zewnętrznych i odbierać komunikaty od nich za pośrednictwem powiązań WebSphere MQ.

Klient zewnętrzny (na przykład portal WWW lub system EIS) może wysyłać komunikat do komponentu SCA w aplikacji za pośrednictwem eksportu lub może być wywoływany przez komponent SCA w aplikacji za pośrednictwem importu.

Powiązanie eksportu WebSphere MQ wdraża komponenty bean sterowane komunikatami (Message Driven Bean - MDB) w celu nasłuchiwania żądań przychodzących do miejsca docelowego receive (odbioru) określonego w powiązaniu eksportu. Miejsce docelowe określone w polu send służy do wysyłania odpowiedzi na żądanie przychodzące, jeśli wywołana aplikacja udostępnia odpowiedź. Dzięki temu klient zewnętrzny może wywoływać aplikacje za pośrednictwem powiązania eksportu.

Importy WebSphere MQ są wiązane z klientami zewnętrznymi i mogą dostarczać do nich komunikaty. Komunikat może wymagać odpowiedzi od klienta zewnętrznego, ale może też nie wymagać tej odpowiedzi.

Więcej informacji na temat interakcji z klientami zewnętrznymi przy użyciu produktu WebSphere MQ można znaleźć w Centrum informacyjnym produktu WebSphere MQ.

Rozwiązywanie problemów z powiązaniem produktu WebSphere MQ:

Istnieje możliwość diagnozowania i naprawiania uszkodzeń oraz warunków niepowodzenia związane z powiązaniem produktu WebSphere MQ.

Podstawowe warunki niepowodzenia

Podstawowe warunki niepowodzenia powiązań produktu WebSphere MQ są określane przez semantykę transakcyjną, przez konfigurację produktu WebSphere MQ lub przez odwołanie do aktualnego działania innych komponentów. Istnieją następujące podstawowe warunki niepowodzenia:

- Niepowodzenie nawiązania połączenia z menedżerem kolejek lub z kolejką produktu WebSphere MQ.
Niepowodzenie nawiązania połączenia z produktem WebSphere MQ w celu odbierania komunikatów powoduje, że nie można uruchomić portu nasłuchiwania MDB. Ten warunek zostaje zarejestrowany w dzienniku produktu WebSphere Application Server. Trwałe komunikaty pozostaną w kolejce produktu WebSphere MQ do czasu ich pomyślnego pobrania (lub do utraty ważności określonej przez produkt WebSphere MQ).
Niepowodzenie nawiązania połączenia z produktem WebSphere MQ w celu wysyłania komunikatów wychodzących powoduje wycofanie transakcji sterującej wysyłaniem.
- Niepowodzenie analizy komunikatu przychodzącego lub budowy komunikatu wychodzącego.
Niepowodzenie w powiązaniu danych powoduje wycofanie transakcji sterującej pracą.
- Niepowodzenie wysłania komunikatu wychodzącego.
Niepowodzenie wysłania komunikatu powoduje wycofanie odpowiedniej transakcji.
- Wiele komunikatów odpowiedzi lub nieoczekiwane komunikaty odpowiedzi.
Element importowany oczekuje tylko jednego komunikatu odpowiedzi na każdy komunikat żądania. Jeśli nadejdzie więcej niż jedna odpowiedź lub nadejdzie odpowiedź spóźniona (czas utraty ważności odpowiedzi SCA minął), zgłoszony zostanie wyjątek środowiska wykonawczego usługi. Transakcja zostanie wycofana, a komunikat odpowiedzi zostanie usunięty z kolejki lub obsłużony przez menedżer zdarzeń zakończonych niepowodzeniem.

Scenariusze niepoprawnego użycia: porównanie z powiązaniem JMS produktu WebSphere MQ

Elementy importowane i eksportowane produktu WebSphere MQ zostały zaprojektowane głównie na potrzeby współpracy z rodzimymi aplikacjami produktu WebSphere MQ i ujawniają mediacjom pełną treść komunikatu produktu WebSphere MQ. Powiązanie JMS produktu WebSphere MQ zostało jednak zaprojektowane z myślą o współpracy z aplikacjami JMS wdrożonymi z produktem WebSphere MQ, który ujawnia komunikaty zgodnie z modelem komunikatów JMS.

W następujących scenariuszach należy stosować powiązania JMS produktu WebSphere MQ zamiast powiązań produktu WebSphere MQ:

- Wywoływanie komponentu bean sterowanego komunikatami JMS (MDB) z modułu SCA, gdy komponent MDB jest wdrożony w dostawcy JMS produktu WebSphere MQ. Należy użyć elementu importowanego JMS produktu WebSphere MQ.

- Umożliwienie wywołania modułu SCA z serwletu komponentów Java EE lub z komponentu EJB za pomocą usług JMS. Należy użyć elementu eksportowanego JMS produktu WebSphere MQ.
- Mediowanie treści komunikatu JMS MapMessage w trakcie przejścia przez produkt WebSphere MQ. Należy użyć elementu eksportowanego i elementu importowanego JMS produktu WebSphere MQ w połączeniu z odpowiednim powiązaniem danych.

Są takie sytuacje, w których powiązanie produktu WebSphere MQ i powiązanie JMS produktu WebSphere MQ JMS powinny współpracować. W szczególności podczas łączenia aplikacji Java EE i innych niż Java EE w produkcie WebSphere MQ należy używać elementu eksportowanego produktu WebSphere MQ oraz elementu importowanego JMS produktu WebSphere MQ (lub odwrotnie) w połączeniu z odpowiednimi powiązaniem danych lub modułami mediacji (bądź dwoma tymi rodzajami obiektów).

Niedostarczone komunikaty

Gdy produkt WebSphere MQ nie może dostarczyć komunikatu do właściwego miejsca docelowego (na przykład z powodu błędów konfiguracji), wysyła go do wyznaczonej kolejki niedostarczonych komunikatów.

Jednocześnie dodaje na początku treści komunikatu nagłówek sygnalizujący niedostarczenie. Ten nagłówek zawiera m.in. informacje o przyczynach niepowodzenia oraz pierwotnym miejscu docelowym.

Komunikaty SCA oparte na produkcie MQ nie są wyświetlane w menedźerze zdarzeń zakończonych niepowodzeniem

Jeśli komunikaty SCA są wynikiem niepowodzenia interakcji produktu WebSphere MQ, powinny znaleźć się w menedźerze zdarzeń zakończonych niepowodzeniem. Jeśli takie komunikaty nie pojawiają się w menedźerze zdarzeń zakończonych niepowodzeniem, należy sprawdzić, czy w bazowym miejscu docelowym produktu WebSphere MQ maksymalną liczbę nieudanych dostaw ustawiono na wartość większą niż 1. Ustawienie wartości 2 lub większej umożliwia interakcję z menedżerem zdarzeń zakończonych niepowodzeniem w trakcie wywołań SCA dotyczących powiązań produktu WebSphere MQ.

Zdarzenia zakończone niepowodzeniem produktu MQ są odtwarzane do niewłaściwego menedżera kolejek

Jeśli na potrzeby połączeń wychodzących ma być używana predefiniowana fabryka połączeń, właściwości połączenia muszą być zgodne z właściwościami zdefiniowanymi w specyfikacji aktywowania używanej na potrzeby połączeń przychodzących.

Predefiniowana fabryka połączeń jest używana do tworzenia połączenia podczas odtwarzania zdarzenia zakończonego niepowodzeniem i dlatego w jej konfiguracji musi być uwzględnione użycie tego samego menedżera kolejek, z którego odebrano pierwotnie komunikat.

Obsługa wyjątków:

Sposób skonfigurowania powiązania określa sposób obsługi wyjątków wywołanych przez procedury obsługi danych lub powiązania danych. Na działanie systemu podczas zgłoszenia takiego wyjątku dodatkowy wpływ ma rodzaj przepływu mediacji.

Jeśli procedura obsługi danych lub powiązanie danych zostanie wywołane przez powiązanie użytkownika, może wystąpić wiele różnych problemów. Procedura obsługi danych może na przykład odebrać komunikat z uszkodzonym ładunkiem lub może próbować odczytać komunikat w niepoprawnym formacie.

Sposób obsługi takiego wyjątku przez powiązanie użytkownika jest określany na podstawie tego, w jaki sposób zaimplementowano procedurę obsługi danych lub powiązanie danych. Zalecanym sposobem utworzenia powiązania danych jest zaprojektowanie go tak, aby zgłaszało wyjątek **DataBindingException**.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, każdy wyjątek procedury obsługi danych jest opakowany w wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

W przypadku wywołania wyjątku czasu wykonywania, między innymi wyjątku **DataBindingException**, mają miejsce następujące działania:

- Jeśli przepływ mediacji jest skonfigurowany jako transakcyjny, komunikat JMS jest domyślnie zapisywany w menedżerze zdarzeń zakończonych niepowodzeniem w celu ręcznego powtórzenia lub usunięcia.

Uwaga: Użytkownik może zmienić tryb odtwarzania dla powiązania tak, aby komunikat był wycofywany zamiast zapisywania go w menedżerze zdarzeń zakończonych niepowodzeniem.

- Jeśli przepływ mediacji nie jest transakcyjny, wyjątek jest rejestrowany, a komunikat zostaje utracony.

W przypadku procedury obsługi danych sytuacja jest podobna. Ponieważ procedura obsługi danych jest wywoływana przez powiązanie danych, wyjątek procedury obsługi danych jest tworzony wewnątrz wyjątku powiązania danych. Dlatego też wyjątek **DataHandlerException** jest zgłaszany użytkownikowi jako wyjątek **DataBindingException**.

Ograniczenia powiązań

W niniejszej sekcji wymieniono ograniczenia korzystania z powiązań.

Ograniczenia powiązania MQ:

Powiązanie MQ charakteryzuje się pewnymi ograniczeniami sposobu używania. Zostały one opisane w niniejszej sekcji.

Brak dystrybucji komunikatów przy użyciu publikowania i subskrybowania

Metoda dystrybucji komunikatów oparta na publikowaniu i subskrybowaniu nie jest aktualnie obsługiwana przez powiązanie MQ, mimo że produkt WMQ obsługuje mechanizm publikowania-subskrybowania. Jednak powiązanie MQ JMS obsługuje tę metodę dystrybucji.

Współużytkowane kolejki odbiorcze

W przypadku wielu powiązań eksportu i importu WebSphere MQ oczekiwane jest, że wszystkie komunikaty obecne w skonfigurowanej dla nich kolejce odbiorczej są skierowane do tego eksportu lub importu. Dlatego też powiązania importu i eksportu należy konfigurować, biorąc pod uwagę następujące kwestie:

- Każdy import MQ musi mieć inną kolejkę odbiorczą, ponieważ powiązanie importu MQ zakłada, że wszystkie komunikaty w kolejce odbiorczej są odpowiedziami na żądania wysłane przez to powiązanie. Jeśli kolejka odbiorcza jest współużytkowana przez więcej niż jeden import, odpowiedzi mogą być odebrane przez niepoprawny import i nie uda się ich korelacja z pierwotnym komunikatem żądania.
- Każdy eksport MQ powinien mieć inną kolejkę odbiorczą, ponieważ w przeciwnym razie nie można przewidzieć, który eksport otrzyma konkretny komunikat żądania.
- Importy i eksporty MQ mogą wskazywać tę samą kolejkę nadawczą.

Ograniczenia powiązań JMS, MQ JMS i ogólnych powiązań JMS:

Powiązania JMS i MQ JMS charakteryzują się pewnymi ograniczeniami.

Konsekwencje generowania powiązań domyślnych

Ograniczenia używania powiązań JMS, MQ JMS i ogólnych powiązań JMS zostały omówione w następujących sekcjach:

- Konsekwencje generowania powiązań domyślnych,
- Schemat korelacji odpowiedzi,
- Obsługa języków dwukierunkowych.

Kiedy użytkownik generuje powiązanie, kilka pól jest wypełnianych wartościami domyślnymi, jeśli nie zostaną wprowadzone w nich żadne wartości. W ten sposób tworzona jest na przykład nazwa fabryki połączeń. Jeśli wiadomo, że aplikacja będzie umieszczana na serwerze, a dostęp do niej będzie uzyskiwany zdalnie za pomocą klienta, na etapie tworzenia powiązania należy wprowadzić nazwy JNDI zamiast przyjmować wartości domyślne, ponieważ jest prawdopodobne, że wartości te będą wymagać kontrolowania za pomocą Konsoli administracyjnej w czasie wykonywania.

Jeśli jednak użytkownik zaakceptuje wartości domyślne, a w późniejszym czasie okaże się, że nie można uzyskać dostępu do aplikacji z poziomu klienta zdalnego, można użyć Konsoli administracyjnej do jawnego ustawienia wartości fabryki połączeń. Należy w tym celu znaleźć pole punktów końcowych dostawcy w ustawieniach fabryki połączeń i dodać wartość, na przykład <nazwa_hosta_serwera>:7276 (jeśli wykorzystywany jest domyślny numer portu).

Schemat korelacji odpowiedzi

Jeśli używany jest schemat korelacji odpowiedzi CorrelationID To CorrelationID, który służy do korelowania komunikatów w operacji żądanie-odpowiedź, w komunikacie należy stosować dynamiczny identyfikator korelacji.

Aby utworzyć dynamiczny identyfikator korelacji w module mediacji przy użyciu edytora przepływów mediacji, należy dodać operację podstawową mediacji Odwzorowanie przed importem zawierającym powiązanie JMS. W tym celu należy najpierw otworzyć edytor odwzorowań. Znane nagłówki architektury SCA będą dostępne w komunikacie docelowym. Następnie należy przeciągnąć pole zawierające unikalny identyfikator w komunikacie źródłowym na identyfikator korelacji w nagłówku JMS komunikatu docelowego.

Obsługa języków dwukierunkowych

W czasie wykonywania w nazwach JNDI (Java Naming and Directory Interface) obsługiwane są wyłącznie znaki ASCII.

Współużytkowane kolejki odbiorcze

W przypadku wielu powiązań eksportu i importu oczekiwane jest, że wszystkie komunikaty obecne w skonfigurowanej dla nich kolejce odbiorczej są skierowane do tego eksportu lub importu. Dlatego też powiązania importu i eksportu należy konfigurować, biorąc pod uwagę następujące kwestie:

- Każde powiązanie importu musi mieć inną kolejkę odbiorczą, ponieważ powiązanie importu zakłada, że wszystkie komunikaty w kolejce odbiorczej są odpowiedziami na żądania wysłane przez to powiązanie. Jeśli kolejka odbiorcza jest współużytkowana przez więcej niż jeden import, odpowiedzi mogą być odebrane przez niepoprawny import i nie uda się ich korelacja z pierwotnym komunikatem żądania.
- Każdy eksport powinien mieć inną kolejkę odbiorczą, ponieważ w przeciwnym razie nie można przewidzieć, który eksport otrzyma konkretny komunikat żądania.
- Importy i eksporty mogą wskazywać tę samą kolejkę nadawczą.

Obiekty biznesowe

W branży oprogramowania komputerowego powstało wiele modeli oprogramowania i środowisk, w których *obiekty biznesowe* stanowią naturalną reprezentację danych biznesowych przetwarzanych przez aplikacje.

Ogólnie, obiekty biznesowe:

- są definiowane przy użyciu standardów branżowych,
- w sposób przezroczysty odwzorowują dane na tabele baz danych lub systemy informacji w przedsiębiorstwach,
- obsługują protokoły zdalnych wywołań,
- udostępniają platformę modelu programowania danych na potrzeby programowania aplikacji.

Komponent Process Designer i produkt Integration Designer udostępniają programistom popularny model obiektów biznesowych do reprezentowania różnych rodzajów jednostek biznesowych z różnych domen.

W produkcie Process Designer obiekty biznesowe koncentrują się na reprezentacji typu danych. Podstawowe obiekty biznesowe (typy zmiennych) są udostępniane w systemowych bibliotekach narzędziowych, a użytkownik może stworzyć niestandardowe typy zmiennych nazywane niestandardowymi obiektami biznesowymi. Więcej informacji na ten temat zawiera sekcja Obiekty biznesowe i zmienne.

W komponencie Integration Designer, który jest dostępny tylko w produkcie IBM Business Process Manager Advanced, obiekty biznesowe mogą reprezentować bardziej skomplikowane konstrukcje XSD. W komponencie Integration Designer obiekty biznesowe są blisko spowinowacone ze schematami XML. Informacje o tym, co należy wziąć pod uwagę podczas integrowania obiektów biznesowych, które zostały zdefiniowane w komponencie Integration Designer przy użyciu obiektów biznesowych zdefiniowanych w produkcie Process Designer zawierają sekcje Zapis lustrzany bibliotek i Nieobsługiwane konstrukcje XML.

W trakcie programowania w komponencie Integration Designer model obiektu biznesowego umożliwia programistom definiowanie obiektów biznesowych jako definicji schematów XML. W czasie wykonywania dane biznesowe zdefiniowane w schematach XML są reprezentowane jako obiekty biznesowe Java. W tym modelu obiekty biznesowe są w mniejszym stopniu oparte na wcześniejszych wersjach roboczych specyfikacji SDO (Service Data Object) i udostępniają kompletny zestaw interfejsów aplikacyjnych modelu programowania wymaganych do manipulowania danymi biznesowymi. Następujące tematy podrzędne zawierają dokładniejsze informacje na temat używania obiektów biznesowych w komponencie Integration Designer.

Definiowanie obiektów biznesowych

Obiekty biznesowe są definiowane przy użyciu edytora obiektów biznesowych w produkcie Integration Designer. Edytor obiektów biznesowych przechowuje obiekty biznesowe jako definicje schematów XML.

Używanie schematu XML do definiowania obiektów biznesowych wiąże się z kilkoma korzyściami:

- Schematy XML udostępniają model definicji danych oparty na standardach oraz platformę współdziałania między odmiennymi, heterogenicznymi systemami i aplikacjami. Schematy XML są używane w połączeniu z językiem WSDL (Web Services Description Language), udostępniając oparte na standardach kontrakty interfejsu między komponentami, aplikacjami i systemami.
- Schematy XML określają rozbudowany model definicji danych służący do reprezentowania danych biznesowych. Ten model zawiera między innymi typy złożone, typy proste, typy zdefiniowane przez użytkownika, dziedziczenie typów oraz licznosc.
- Obiekty biznesowe mogą być definiowane na podstawie interfejsów biznesowych oraz danych zdefiniowanych w języku WSDL (Web Services Description Language), a także na podstawie schematów XML pochodzących od organizacji nadzorujących standardy branżowe lub z innych systemów i aplikacji. Integration Designer może bezpośrednio importować te obiekty biznesowe.

Integration Designer udostępnia również obsługę wykrywania danych biznesowych w bazach danych oraz korporacyjnych systemach informacyjnych, a także umożliwia generowanie definicji obiektów biznesowych w postaci standardowych schematów XML dla tych danych biznesowych. Obiekty biznesowe wygenerowane w ten sposób są często nazywane *obiettami biznesowymi specyficznymi dla aplikacji*, ponieważ odzwierciedlają strukturę danych biznesowych zdefiniowanych w korporacyjnym systemie informacyjnym.

Jeśli proces manipuluje danymi z wielu różnych systemów informacyjnych, warto przetransformować odmienne reprezentacje danych biznesowych (na przykład Klient_EIS1 i Klient_EIS2 lub Zamówienie_EIS1 i Zamówienie_EIS2) na pojedynczą reprezentację kanoniczną (na przykład Klient lub Zamówienie). Reprezentacja kanoniczna jest często nazywana *ogólnym obiectem biznesowym*.

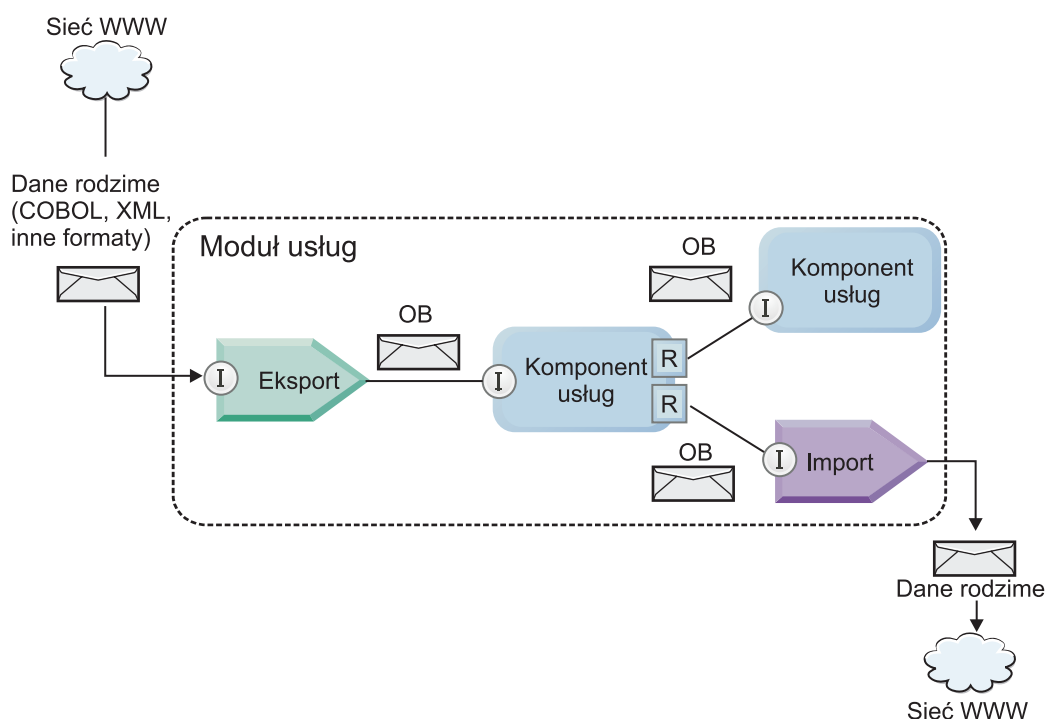
Definicje obiektów biznesowych, szczególnie w przypadku ogólnych obiektów biznesowych, są często używane przez więcej niż jedną aplikację. Aby umożliwić to ponowne wykorzystywanie, produkt Integration Designer pozwala tworzyć obiekty biznesowe w bibliotekach, które można następnie powiązać z wieloma modułami aplikacji.

Zadania dla usług udostępnianych przez moduł aplikacji SCA (Service Component Architecture) i konsumowanych przez ten moduł, jak również zadania używane do tworzenia komponentów w module aplikacji, są definiowane przy użyciu języka WSDL (Web Services Description Language). W zadaniu dokument WSDL może reprezentować zarówno operacje, jak i obiekty biznesowe, które są definiowane przez schemat XML w celu reprezentowania danych biznesowych.

Praca z obiektami biznesowymi

Architektura SCA (Service Component Architecture) udostępnia środowisko do definiowania modułu aplikacji, usługi do jego udostępniania i wykorzystywania oraz zestaw komponentów udostępniających jego logikę biznesową. Obiekty biznesowe pełnią w aplikacji ważną rolę: służą do definiowania danych biznesowych używanych do opisu zadań usług i komponentów oraz danych biznesowych przetwarzanych przez komponenty.

Na następującym diagramie przedstawiono moduł aplikacji SCA i pokazano wiele miejsc, w których programista pracuje z obiektami biznesowymi.



Rysunek 86. Obiekty biznesowe jako reprezentacja danych przekazywanych między usługami w aplikacji

Uwaga: W tym temacie opisano, w jaki sposób obiekty biznesowe są używane w modułach aplikacji SCA. Jeśli programista używa interfejsów Java, w modułach aplikacji SCA może przetwarzać również obiekty Java.

Model programowania obiektów biznesowych

Model programowania obiektów biznesowych składa się z zestawu interfejsów Java, które reprezentują:

- definicję obiektu biznesowego i dane instancji,
- zestaw usług do obsługi operacji na obiektach biznesowych.

Definicje typów obiektów biznesowych są reprezentowane przez interfejsy `commonj.sdo.Type` i `commonj.sdo.Property`. Model programowania obiektów biznesowych udostępnia zestaw reguł odwzorowywania informacji zawartych w typach złożonych schematu XML na interfejs `Type` oraz odwzorowywania wszystkich elementów z definicji typu złożonego na interfejs `Property`.

Instancje obiektów biznesowych są reprezentowane przez interfejs `commonj.sdo.DataObject`. Model programowania obiektów biznesowych jest beztypowy; oznacza to, że ten sam interfejs `commonj.sdo.DataObject` może być używany do reprezentowania różnych definicji obiektów biznesowych, takich jak Klient i Zamówienie. O tym, które właściwości mogą być ustawiane i pobierane z każdego obiektu biznesowego, decyduje typ informacji zdefiniowanych w schemacie XML powiązany z każdym obiektem biznesowym.

Zachowanie modelu programowania obiektów biznesowych jest oparte na specyfikacji Service Data Object 2.1. Dodatkowe informacje można znaleźć w specyfikacji SDO 2.1 for Java oraz w podręcznikach i w dokumentacji języka Java pod adresem: <http://www.oasis-openca.org/>.

Usługi obiektów biznesowych obsługują operacje na różnych etapach cyklu życia obiektów biznesowych, takie jak tworzenie, porównywanie, analizowanie czy serializacja obiektów biznesowych.

Specyfikację modelu programowania obiektów biznesowych można znaleźć w sekcjach Programowanie korzystania z usług obiektów biznesowych i Dokumentacja wygenerowanych interfejsów API i interfejsów SPI dla obiektów biznesowych.

Powiązania, powiązania danych i procedury obsługi danych

Rys. 41 na stronie 151 przedstawia, w jaki sposób dane biznesowe używane podczas wywoływania usług udostępnianych przez moduły aplikacji SCA są transformowane w obiekty biznesowe, aby umożliwić manipulowanie nimi w komponentach SCA. Podobnie, obiekty biznesowe przetwarzane w komponentach SCA są przekształcane w format danych wymagany przez usługi zewnętrzne.

W niektórych przypadkach, na przykład podczas wiązania usługi Web Service, powiązanie używane w usługach eksportu i importu automatycznie transformuje dane do odpowiedniego formatu. W innych przypadkach, na przykład podczas wiązania JMS, programiści mogą udostępnić powiązanie danych lub procedurę obsługi danych, która przekształca formaty inne niż rodzime w obiekty biznesowe reprezentowane przez interfejs `DataObject`.

Więcej informacji na temat programowania powiązań danych i procedur obsługi danych można znaleźć w sekcjach “Procedury obsługi danych” na stronie 57 oraz “Powiązania danych” na stronie 59.

Komponenty

Komponenty SCA zawierają definicje zadań usług udostępniania i wykorzystywania przy użyciu języka WSDL (Web Services Description Language) i schematu XML. Dane biznesowe przekazywane w środowisku SCA między komponentami są reprezentowane przez obiekty biznesowe przy użyciu interfejsu `DataObject`. W środowisku SCA następuje sprawdzenie, czy typy obiektów biznesowych są zgodne z zadaniem interfejsu zdefiniowanym w wywołanym komponentcie.

Abstrakcje modelu programowania do manipulowania obiektami biznesowymi są różne w różnych komponentach. Komponent POJO i operacja podstawowa Custom w komponentcie przepływu mediacji umożliwiają bezpośrednie manipulowanie obiektami biznesowymi, zezwalając na używanie języka Java bezpośrednio w usługach i interfejsach programowania obiektów biznesowych. Większość komponentów udostępnia abstrakcje wyższego poziomu do manipulowania obiektami biznesowymi, ale dostępne są także fragmenty kodu w języku Java do definiowania niestandardowego zachowania usług i interfejsów obiektów biznesowych.

Obiekty biznesowe można transformować przy użyciu kombinacji komponentów Interface Flow Mediation i Business Object Map albo przy użyciu komponentu przepływu mediacji i jego operacji podstawowej XML Map. Te możliwości transformacji obiektów biznesowych są użyteczne do przekształcania obiektów biznesowych specyficznych dla aplikacji w ogólne obiekty biznesowe i odwrotnie.

Specjalne obiekty biznesowe

Obiekty SMO i grafy biznesowe to dwa specjalne typy obiektów biznesowych, używane w aplikacjach do specjalnych celów.

Obiekt SMO

Obiekt SMO (Service Message Object) jest to specjalny obiekt biznesowy używany przez komponenty przepływu mediacji do reprezentowania kolekcji danych powiązanych z wywołaniem usługi.

Obiekt SMO ma stałą strukturę najwyższego poziomu, która składa się z nagłówek, kontekstu, treści i załączników (jeśli istnieją).

- W nagłówkach są przechowywane informacje związane z wywołaniem usługi poprzez konkretny protokół lub powiązanie. Przykładowo są to nagłówki SOAP i nagłówki JMS.
- Dane kontekstowe to dodatkowe informacje logicznie powiązane z wywołaniem, które jest przetwarzane przez komponent przepływu mediacji. Zwykle te informacje nie są częścią danych aplikacji wysyłanych lub otrzymywanych przez klienty.
- Treść obiektu SMO to ładunek danych biznesowych, który reprezentuje zasadniczy komunikat aplikacji lub dane wywołania w formie standardowego obiektu biznesowego.

Obiekt SMO może zawierać również załączniki umożliwiające wywoływanie usług Web Services przy użyciu nagłówek SOAP z załącznikami.

W przepływach mediacji wykonywane są takie zadania, jak żądania przepływu i transformacja danych, a obiekt SMO zapewnia złożony widok nagłówek i treści ładunku w jednej ujednoczonej strukturze.

Graf biznesowy

Graf biznesowy to specjalny obiekt biznesowy używany do obsługi synchronizacji danych w scenariuszach integracji.

Rozważmy przykład, w którym w dwóch korporacyjnych systemach informacyjnych istnieją reprezentacje konkretnej kolejności. Gdy kolejność w jednym systemie ulegnie zmianie, do drugiego systemu może zostać wysłany komunikat informujący o potrzebie synchronizacji kolejności danych. Grafy biznesowe obsługują przesyłanie do drugiego systemu zmienionej porcji danych oraz informacji podsumowujących zmiany i definiujących ich typ.

W tym przykładzie graf biznesowy Kolejność mógłby przekazać do innego systemu informację, że jeden z wierszy w kolejności został usunięty i że właściwość wysyłania danych została zaktualizowana.

Grafy biznesowe można łatwo dodawać do istniejących obiektów biznesowych, korzystając z programu Integration Designer. Występują one najczęściej w scenariuszach, w których są używane adaptory WebSphere, oraz w scenariuszach obsługi migracji aplikacji serwera WebSphere InterChange Server.

Tryb analizowania obiektu biznesowego

Produkt Integration Designer udostępnia właściwość dla modułów lub bibliotek, której można użyć do zmiany trybu analizowania XML dla obiektów biznesowych na przyspieszony lub opóźniony.

- Jeśli wybrana jest opcja *przyspieszony*, strumień bajtów XML są analizowane w trybie przyspieszonym w celu utworzenia obiektu biznesowego.
- Jeśli wybrana jest opcja *opóźniony*, obiekt biznesowy jest tworzony w normalny sposób, ale rzeczywiste analizowanie strumienia bajtów XML jest odraczane i realizowane częściowo dopiero podczas uzyskiwania dostępu do właściwości obiektu biznesowego.

W obu trybach analizowania XML dane inne niż XML są zawsze analizowane w trybie przyspieszonym w celu utworzenia obiektu biznesowego.

Uwagi na temat wybierania trybu analizowania obiektu biznesowego

Tryb analizowania obiektu biznesowego określa sposób analizowania danych XML w czasie wykonywania. Tryb analizowania obiektu biznesowego jest definiowany w module lub w bibliotece podczas ich tworzenia. Tryb analizowania dla modułu lub biblioteki można zmienić, ale należy zdawać sobie sprawę z konsekwencji.

Tryb analizowania obiektu biznesowego jest ustawiany na poziomie modułu i biblioteki. Moduły, które zostały utworzone w wersji produktu IBM Integration Designer wcześniejszej niż wersja 7, będą uruchamiane w przyspieszonym trybie analizowania bez konieczności wprowadzania zmian. Moduły i biblioteki utworzone w produkcie IBM Integration Designer w wersji 7 lub późniejszej domyślnie otrzymają najodpowiedniejszy tryb analizowania, w zależności od kilku czynników takich jak tryb analizowania istniejących projektów w obszarze roboczym lub tryb analizowania projektów zależnych albo innych projektów zawartych w tym samym rozwiązaniu itp. Tryb analizowania obiektu biznesowego dla modułu lub biblioteki można zmienić tak, aby był odpowiedni dla implementacji, ale należy uwzględnić następujące uwagi.

Uwagi

- Opóźniony tryb analizowania obiektu biznesowego przetwarza dane XML szybciej. Występują jednak różnice dotyczące zgodności między trybem przyspieszonym i trybem opóźnionym, z których należy zdawać sobie sprawę przed dokonaniem zmiany konfiguracji modułu lub biblioteki. Te różnice będą miały wpływ na działanie modułów w czasie wykonywania. Informacje o tym, który tryb analizowania jest optymalny dla danej aplikacji, zawiera temat **Zalety stosowania opóźnionego (zamiast przyspieszonego) trybu analizowania** w sekcji stron pokrewnych.
- Moduł można skonfigurować do działania tylko w jednym trybie analizowania. Biblioteki można skonfigurować tak, aby obsługiwały jeden z dostępnych trybów analizowania lub oba z nich. Do biblioteki skonfigurowanej tak, aby obsługiwała oba tryby analizowania może odwoływać się zarówno moduł używający przyspieszonego trybu analizowania, jak i moduł używający opóźnionego trybu analizowania. Tryb analizowania biblioteki stosowany w czasie wykonywania jest określany przez moduły odwołujące się do tej biblioteki. W czasie wykonywania moduł deklaruje swój tryb analizowania. Ten tryb analizowania jest następnie używany przez moduł oraz wszystkie biblioteki używane przez ten moduł.
- Moduły i biblioteki skonfigurowane dla różnych trybów analizowania są zgodne w następujących przypadkach:
 - Moduły i biblioteki skonfigurowane z opóźnionym trybem analizowania są zgodne z bibliotekami, które używają opóźnionego trybu analizowania lub zarówno przyspieszonego jak i opóźnionego trybu analizowania.
 - Moduły i biblioteki skonfigurowane z przyspieszonym trybem analizowania są zgodne z bibliotekami, które używają przyspieszonego trybu analizowania lub zarówno przyspieszonego jak i opóźnionego trybu analizowania.
 - Biblioteki skonfigurowane z opóźnionym i przyspieszonym trybem analizowania są zgodne tylko z bibliotekami, które używają zarówno opóźnionego, jak i przyspieszonego trybu analizowania.
- Dla współdziałających modułów komunikujących się przy użyciu powiązania SCA należy użyć tego samego trybu analizowania. Jeśli moduły komunikują się przy użyciu różnych trybów analizowania, mogą wystąpić problemy związane z wydajnością.

Pojęcia pokrewne:

“Zalety stosowania opóźnionego (zamiast przyspieszonego) trybu analizowania” na stronie 154

W jednych aplikacjach za korzystniejszy uznaje się opóźniony tryb analizowania XML, w innych lepszą wydajność zapewnia stosowanie przyspieszonego trybu analizowania. Zaleca się wykonanie testu porównawczego obu trybów analizowania i na tej podstawie określenie, który tryb jest bardziej odpowiedni dla konkretnej aplikacji.

Zalety stosowania opóźnionego (zamiast przyspieszonego) trybu analizowania

W jednych aplikacjach za korzystniejszy uznaje się opóźniony tryb analizowania XML, w innych lepszą wydajność zapewnia stosowanie przyspieszonego trybu analizowania. Zaleca się wykonanie testu porównawczego obu trybów analizowania i na tej podstawie określenie, który tryb jest bardziej odpowiedni dla konkretnej aplikacji.

Aplikacje, w których są analizowane duże strumienie danych XML, prawdopodobnie będą miały lepszą wydajność, gdy będzie używany opóźniony tryb analizowania XML. Poprawa wydajności będzie większa, gdy strumień bajtów XML będzie wzrastać, a ilość danych w strumieniu bajtów, do którego aplikacja uzyskuje dostęp, będzie maleć.

Następujące aplikacje prawdopodobnie będą osiągały lepszą wydajność przy użyciu przyspieszonego trybu analizowania:

- Aplikacje, w których są analizowane strumienie danych innych niż XML.
- Aplikacje, które korzystają z komunikatów utworzonych przy użyciu usługi BOFactory.
- Aplikacje, w których są analizowane bardzo małe komunikaty XML.

Odsyłacze pokrewne:

“Uwagi na temat wybierania trybu analizowania obiektu biznesowego” na stronie 153

Tryb analizowania obiektu biznesowego określa sposób analizowania danych XML w czasie wykonywania. Tryb analizowania obiektu biznesowego jest definiowany w module lub w bibliotece podczas ich tworzenia. Tryb analizowania dla modułu lub biblioteki można zmienić, ale należy zdawać sobie sprawę z konsekwencji.

Uwagi na temat migrowania i programowania aplikacji

Podczas konfigurowania aplikacji, która została napisana pod kątem stosowania przyspieszonego trybu analizowania, w przypadku korzystania z opóźnionego trybu analizowania lub podczas planowania przełączania aplikacji między opóźnionym i przyspieszonym trybem analizowania należy pamiętać o różnicach między trybami oraz o zagadnieniach związanych z przełączaniem trybów.

Obsługa błędów

Jeśli strumień bajtów XML jest uszkodzony, wystąpią wyjątki analizowania.

- W przyspieszonym trybie analizowania XML wyjątki są generowane w trakcie analizowania obiektu biznesowego z przychodzącego strumienia danych XML.
- Jeśli zostanie skonfigurowany opóźniony tryb analizowania XML, wyjątki analizowania wystąpią z pewnym opóźnieniem — po uzyskaniu dostępu do właściwości obiektów biznesowych i po zakończeniu analizowania błędnej porcji danych XML.

W celu obsługi błędnych danych XML należy wybrać jedną z następujących opcji:

- Wdrożenie magistrali komunikacyjnej na krawędziach, aby sprawdzać poprawność przychodzących plików XML.
- Utworzenie logiki opóźnionego wykrywania błędów w punkcie, w którym są dostępne właściwości obiektów biznesowych.

Stosy wyjątków i komunikaty

Ponieważ przyspieszony i opóźniony tryb analizowania XML mają różne implementacje bazowe, stosy wywołań zgłaszane przez usługi i interfejsy programistyczne obiektów biznesowych mają wprawdzie taką samą nazwę klasy wyjątków, ale mogą nie zawierać takiego samego komunikatu o wyjątku lub opakowanego zestawu klas wyjątków przeznaczonych dla konkretnej implementacji.

Format serializacji XML

W opóźnionym trybie analizowania XML optymalizowanie wydajności podczas serializacji polega na próbach kopiowania niezmiennych fragmentów XML z przychodzącego strumienia bajtów do strumienia wychodzącego. Powoduje to wzrost wydajności, ale format serializacji wychodzącego strumienia bajtów XML może zostać zmieniony, jeśli cały obiekt biznesowy został zaktualizowany w opóźnionym trybie analizowania XML lub gdy był uruchomiony w przyspieszonym trybie analizowania XML.

Nawet jeśli format serializacji XML nie będzie dokładnie taki sam pod względem syntaktycznym, wartość semantyczna udostępniana przez obiekt biznesowy będzie dokładnie taka sama, niezależnie od użytych trybów analizowania, a strumień XML można bezpiecznie przekazywać między aplikacjami używającymi różnych trybów analizowania.

Analizator poprawności instancji obiektów biznesowych

Analizator poprawności instancji obiektów biznesowych dla opóźnionego trybu analizowania XML udostępnia dokładniejsze sprawdzanie poprawności obiektów biznesowych, a szczególnie sprawdzanie poprawności aspektów wartości właściwości. Dzięki temu analizator poprawności instancji obiektów biznesowych dla opóźnionego trybu analizowania wychwytuje dodatkowe problemy w porównaniu z analizatorem dla przyspieszonego trybu analizowania oraz udostępnia bardziej szczegółowe komunikaty o błędach.

Odwzorowania w wersji 602 języka XML

Przepływy mediacji, które zostały pierwotnie utworzone w programie WebSphere Integration Developer w wersji wcześniejszej niż 6.1, mogą zawierać operacje podstawowe Odwzorowanie używające odwzorowania lub arkusza stylów, których nie można bezpośrednio uruchamiać w opóźnionym trybie analizowania XML. Podczas migrowania aplikacji, aby umożliwić stosowanie opóźnionego trybu analizowania XML, pliki odwzorowania powiązane z operacjami podstawowymi Odwzorowanie mogą zostać automatycznie zaktualizowane przez kreator migracji do działania w nowym trybie. Jeśli jednak operacja podstawowa Odwzorowanie odwołuje się bezpośrednio do arkusza stylów, który był edytowany ręcznie, arkusz stylów nie zostanie poddany migracji i nie będzie można go uruchomić w opóźnionym trybie analizowania XML.

Prywatne niepublikowane interfejsy API

Jeśli w aplikacji są wykorzystywane niepublikowane, prywatne interfejsy programistyczne obiektów biznesowych przeznaczone dla konkretnej implementacji, wówczas prawdopodobnie po przełączeniu trybu analizowania kompilowanie aplikacji nie powiedzie się. W przyspieszonym trybie analizowania te prywatne interfejsy są zwykle klasami implementacji obiektów biznesowych zdefiniowanymi w środowisku EMF (Eclipse Modeling Framework).

We wszystkich przypadkach usilnie zaleca się usunięcie prywatnych interfejsów API z aplikacji.

Interfejsy API dla obiektów SMO w środowisku EMF

Komponent mediacji w programie IBM Integration Designer udostępnia możliwość manipulowania zawartością komunikatów przy użyciu klas w języku Java oraz interfejsów dostarczanych w pakiecie `com.ibm.websphere.sibx.smobo`. W opóźnionym trybie analizowania XML można używać interfejsów w języku Java z pakietu `com.ibm.websphere.sibx.smobo`, jednak metody, które odwołują się bezpośrednio do interfejsów i klas EMF (Eclipse Modeling Framework) lub które są dziedziczone z interfejsów EMF, prawdopodobnie nie powiodą się.

W opóźnionym trybie analizowania XML obiektu `ServiceMessageObject` i jego treści nie można rzutować na obiekty EMF.

Usługa BOMode

Usługa BOMode służy do określania, czy bieżącym trybem analizowania XML jest tryb opóźniony czy przyspieszony.

Migracja

Wszystkie aplikacje w wersjach wcześniejszych niż wersja 7.0.0.0 działają w przyspieszonym trybie analizowania XML. Jeśli zostały poddane migracji podczas wykonywania przez narzędzia do migracji BPM w czasie wykonywania, w dalszym ciągu będą działać w przyspieszonym trybie analizowania XML.

Aby można było skonfigurować aplikację w wersji wcześniejszej niż wersja 7.0.0.0 do używania opóźnionego trybu analizowania XML, najpierw należy użyć programu Integration Designer i poddać migracji artefakty aplikacji. Po zakończeniu migracji można skonfigurować aplikację do używania opóźnionego trybu analizowania XML.

Informacje o migrowaniu artefaktów w programie Integration Designer zawiera temat Migrowanie artefaktów źródłowych, a informacje o ustawianiu trybu analizowania można znaleźć w temacie Konfigurowanie trybu analizowania obiektów biznesowych dla modułów i bibliotek.

Relacje

Relacja jest powiązaniem między co najmniej dwoma obiektami danych, zazwyczaj obiektami biznesowymi. W produkcie IBM Business Process Manager Advanced relacje można wykorzystywać do transformowania danych, które są równoważne w różnych obiektach biznesowych i w ramach różnych danych, ale są reprezentowane w odmienny sposób. Można ich również użyć do tworzenia powiązań między różnymi obiektami znajdującymi się w różnych aplikacjach. Relacje można współużytkować w aplikacjach, w różnych rozwiązaniach, a nawet w różnych produktach.

Usługa relacji w produkcie IBM Business Process Manager Advanced udostępnia infrastrukturę oraz operacje służące do zarządzania relacjami. Ponieważ umożliwia ona obsługę obiektów biznesowych niezależnie od tego, gdzie znajdują się te obiekty, może ona udostępniać całościowy, ujednoczony widok wszystkich aplikacji w przedsiębiorstwie i może służyć jako element budowania rozwiązań do zarządzania procesami biznesowymi (BPM). Ze względu na fakt, że relacje są rozszerzalne i mogą podlegać zarządzaniu, istnieje możliwość używania ich w złożonych rozwiązaniach integracji.

Czym są relacje?

Relacja jest powiązaniem między obiektami biznesowymi. Każdy obiekt biznesowy w relacji jest nazywany *uczestnikiem* relacji. Każdy uczestnik relacji jest wyróżniany od innych uczestników na podstawie funkcji lub *roli*, jaką pełni w tej relacji. Relacja zawiera listę ról.

Definicja relacji opisuje wszystkie role oraz powiązania między nimi. Opisuje również ogólny „kształt” relacji. Jedna rola może mieć na przykład tylko jednego uczestnika, ale inna rola może mieć dowolną liczbę uczestników. Użytkownik może na przykład zdefiniować relację *samochód-właściciel*, w której wiele samochodów może mieć jednego właściciela. Jedna instancja może mieć przykładowo następujących uczestników dla każdej z tych ról:

- Samochód (Ferrari)
- Właściciel (Jan)

Definicja relacji jest szablonem dla *instancji* relacji. Instancja jest obiektem czasu wykonywania dla relacji. W przykładzie *samochód-właściciel* instancja może opisywać dowolne spośród poniższych powiązań:

- Jan jest właścicielem samochodu Ferrari
- Anna jest właścicielem samochodu Mazda
- Tomasz jest właścicielem samochodu Ferrari

Dzięki relacjom użytkownik nie musi budować niestandardowych trwałych mechanizmów śledzenia relacji w ramach logiki biznesowej. W niektórych scenariuszach usługa relacji realizuje całość pracy za użytkownika. Patrzy przykład opisany w sekcji Relacje tożsamościowe.

Scenariusze

Poniżej przedstawiono typowy przykład sytuacji, w której rozwiązanie integracji może wykorzystywać relacje. Duża korporacja dokonuje zakupu wielu form lub jednostek organizacyjnych. Każda jednostka organizacyjna używa innego oprogramowania do monitorowania personelu i komputerów notebook. Firma wymaga sposobu monitorowania jej pracowników oraz ich komputerów notebook. Potrzebne jest rozwiązanie, które umożliwi realizowanie następujących działań:

- Wyświetlanie wszystkich pracowników w poszczególnych jednostkach organizacyjnych tak, jakby znajdowali się w jednej bazie danych.
- Wyświetlanie pojedynczego widoku wszystkich komputerów notebook używanych przez tych pracowników.
- Zezwolenie pracownikom na logowanie się w systemie i dokonanie zakupu komputera notebook.
- Obsługa różnych systemów aplikacji korporacyjnych w poszczególnych jednostkach organizacyjnych.

Aby to zrealizować, firma potrzebuje mechanizmu, który zagwarantuje że Jan Kowalski i Jan Krzysztof Kowalski w różnych aplikacjach są postrzegani jako ten sam pracownik. Niezbędny jest na przykład sposób konsolidowania pojedynczego obiektu w wielu obszarach aplikacji.

Bardziej złożone scenariusze relacji obejmują budowanie procesów BPEL, które uwzględniają relacje między różnymi obiektami w różnych aplikacjach. W złożonych scenariuszach relacji obiekty biznesowe znajdują się w rozwiązaniu integracji, a nie w aplikacjach. Usługa relacji udostępnia platformę do trwałego zarządzania relacjami. Przed wprowadzeniem usługi relacji użytkownik musiał budować własną usługę utrwalania obiektów. Poniżej przedstawiono dwa przykłady złożonych scenariuszy relacji:

- W aplikacji SAP istnieje obiekt biznesowy **samochód** z numerem nadwozia, a użytkownik chce śledzić, że ten samochód jest własnością określonej osoby. Jednak relacja własności dotyczy osoby, informacje o której są umieszczone w aplikacji PeopleSoft. W tym wzorcu relacji istnieją dwa rozwiązania i konieczne jest utworzenie między nimi mostu.
- Duża firma handlu detalicznego chce monitorować towary zwracane w celu uzyskania zwrotu gotówki lub kredytu. W tym procesie są uwzględniane dwie aplikacje: system zarządzania zamówieniami do obsługi zakupów oraz system obsługi zwrotów do obsługi zwrotów. Obiekty biznesowe znajdują się w więcej niż jednej aplikacji i konieczny jest sposób wyświetlania relacji między nimi.

Wspólne wzorce użycia

Najpopularniejszymi wzorcami relacji są wzorce *równoważności*. Są one oparte na odwołaniach lub na korelacji. Istnieją dwa typy relacji, które są zgodne z tym wzorcem: *nietożsamościowe* i *tożsamościowe*.

- **Relacje nietożsamościowe** ustanawiają powiązania między obiektami biznesowymi lub innymi danymi w konfiguracji jeden-do-wielu lub wiele-do-wielu. Dla każdej instancji relacji może istnieć jedna lub więcej instancji każdego uczestnika. Jednym z typów relacji nietożsamościowych jest statyczna relacja słownikowa. Jej przykładem może być relacja, w której kod CA w aplikacji SAP jest powiązany z kodem **Kalifornia** w aplikacji Siebel.

•

Relacje tożsamościowe ustanawiają powiązania między obiektami biznesowymi lub innymi danymi w konfiguracji jeden-do-jednego. Dla każdej instancji relacji może istnieć tylko jedna instancja każdego uczestnika. Relacje tożsamościowe przechwytyją odwołania między obiektami biznesowymi, które są semantycznie równoważne, ale są identyfikowane w różny sposób w poszczególnych aplikacjach. Każdy uczestnik relacji jest powiązany z obiektem biznesowym, który ma wartość (lub kombinację wartości) unikalnie identyfikującą obiekt. Relacje tożsamościowe zwykle transformują atrybuty kluczowe obiektów biznesowych, takie jak identyfikatory i kody produktów.

Jeśli na przykład w aplikacjach SAP, PeopleSoft i Siebel istnieje obiekt biznesowy **samochód** należy zbudować rozwiązanie, które synchronizuje je ze sobą, zazwyczaj konieczne byłoby wprowadzenie ręcznie utworzonej logiki synchronizacji relacji w postaci sześciu odwzorowań:

SAP -> ogólne

ogólne -> SAP

PeopleSoft-> ogólne

ogólne -> PeopleSoft

Siebel -> ogólne

ogólne -> Siebel

Jeśli jednak w rozwiązaniu wykorzystywane są relacje, usługa relacji udostępnia wstępnie zbudowane implementacje wzorca, które utrzymują dla użytkownika wszystkie te instancje relacji.

Narzędzia do pracy z relacjami

Edytor relacji w produkcie Integration Designer jest narzędziem służącym do modelowania i projektowania relacji oraz ról integracji biznesowej. Szczegółowe informacje uzupełniające oraz informacje o czynnościach dotyczące tworzenia relacji i używania edytora relacji zawiera sekcja Tworzenie relacji.

Usługa relacji jest usługą infrastruktury w produkcie IBM Business Process Manager, która utrzymuje relacje i role w systemie oraz udostępnia operacje służące do zarządzania relacjami i rolami.

Menedżer relacji jest interfejsem administracyjnym służącym do zarządzania relacjami. Dostęp do niego można uzyskać za pośrednictwem stron menedżera relacji w Konsoli administracyjnej.

Relacje można wywoływać programowo za pośrednictwem funkcji API usługi relacji.

Usługa relacji

Usługa relacji zapisuje dane relacji w tabelach relacji, w których śledzi ona wartości specyficzne dla aplikacji w poszczególnych aplikacjach i w różnych rozwiązaniach. Usługa relacji udostępnia operacja do zarządzania relacjami i rolami.

Jak działają relacje

Relacje i role są definiowane przy użyciu graficznego interfejsu narzędzia edytora relacji w produkcie Integration Designer. Usługa relacji zapisuje dane korelacji w tabelach w bazie danych relacji w domyślnym źródle danych określanym podczas konfigurowania usługi relacji. W oddzielnej tabeli (nazywanej czasem tabelą uczestników) zapisywane są informacje na temat każdego uczestnika w relacji. Usługa relacji używa tych tabel relacji do śledzenia odpowiednich wartości specyficznych dla aplikacji oraz do propagowania zaktualizowanych informacji do wszystkich rozwiązań.

Relacje, które są artefaktami biznesowymi, są wdrażane w projekcie lub w bibliotece współużytkowanej. Podczas pierwszego wdrożenia usługa relacji wypełnia dane.

Instancje relacji są aktualizowane lub pobierane w czasie wykonywania, kiedy odwzorowania lub inne komponenty produktu IBM Business Process Manager wymagają instancji relacji. Jest to zależne od scenariusza.

Dane instancji relacji i roli mogą być manipulowane przy użyciu trzech sposobów:

- Wywołań fragmentów kodu Java funkcji API usługi relacji komponentu produktu IBM Business Process Manager.
- Transformacji relacji w usłudze odwzorowywania obiektów biznesowych produktu IBM Business Process Manager.
- Przy użyciu narzędzia menedżera relacji.

Szczegółowe informacje uzupełniające oraz informacje o czynnościach dotyczące tworzenia relacji, identyfikowania typów relacji oraz używania edytora relacji zawiera temat Tworzenie relacji.

Menedżer relacji

Menedżer relacji jest interfejsem administracyjnym służącym do zarządzania relacjami. Dostęp do niego można uzyskać za pośrednictwem stron menedżera relacji w Konsoli administracyjnej.

Menedżer relacji udostępnia graficzny interfejs użytkownika służący do tworzenia danych relacji i ról oraz manipulowania nimi w czasie wykonywania. Jednostkami relacji można zarządzać na wszystkich poziomach: instancji relacji, instancji roli oraz danych atrybutu i danych właściwości. Za pomocą menedżera relacji można realizować następujące działania:

- Wyświetlanie listy relacji w systemie oraz szczegółowych informacji na temat poszczególnych relacji.
- Zarządzanie instancjami relacji:
 - Realizowanie zapytań o dane relacji w celu wyświetlenia podzbiorów danych instancji.
 - Realizowanie zapytań o dane relacji w celu wyświetlenia podzbiorów danych instancji przy użyciu widoków bazy danych.
 - Wyświetlanie listy instancji relacji, które są zgodne z zapytaniem o relację, a także szczegółowych informacji na temat instancji.
 - Edytowanie wartości właściwości dla instancji relacji.
 - Tworzenie i usuwanie instancji relacji.
- Zarządzanie rolami i instancjami roli:
 - Wyświetlanie szczegółów roli lub instancji roli.
 - Edytowanie właściwości instancji roli.
 - Tworzenie i usuwanie instancji roli dla relacji.
 - Wycofywanie danych instancji relacji do punktu w czasie, w którym jest pewne, że są one poprawne.

- Importowanie danych z istniejącej relacji statycznej do systemu lub eksportowanie danych z istniejącej relacji statycznej do pliku RI lub CSV.
- Usuwanie schematów i danych relacji z repozytorium po zdeinstalowaniu aplikacji wykorzystującej te schematy i dane.

Relacje w środowiskach wdrożenia sieciowego

W środowiskach wdrożenia sieciowego można używać relacji bez konieczności dodatkowego konfigurowania.

W środowiskach wdrożenia sieciowego relacje są instalowane w klastrze aplikacji. Relacje są następnie widoczne w ramach klastra, a wszystkie serwery w klastrze mają dostęp do danych instancji przechowywanych w bazie danych relacji. Możliwość uruchamiania usługi relacji w środowisku wdrożenia sieciowego powoduje, że środowisko to charakteryzuje się skalowalnością i wysoką dostępnością.

Menedżer relacji umożliwia zarządzanie relacjami w różnych klastrach za pośrednictwem scentralizowanego interfejsu administracyjnego. Połączenie menedżera relacji z serwerem w klastrze jest nawiązywane poprzez wybór komponentu MBean relacji tego serwera.

Funkcje API usługi relacji

Relacje można wywoływać programowo przy użyciu funkcji API usługi relacji. Mogą być one wywoływane wewnątrz odzworowań obiektów biznesowych lub poza nimi.

Dostępne są trzy typy funkcji API:

- Funkcje API do manipulowania instancjami relacji (między innymi do bezpośredniego tworzenia, aktualizowania i usuwania danych instancji).
- Funkcje API do obsługi wzorców relacji (między innymi `correlate()`, `correlateforeignKeyLookup`).
- Wzorce wyszukiwania relacji (funkcje API wyszukiwania).

Magistrala usług korporacyjnych w produkcie IBM Business Process Manager

Produkt IBM Business Process Manager obsługuje integrowanie usług aplikacji, obejmujące takie same możliwości jak produkt WebSphere Enterprise Service Bus.

Łączenie usług za pośrednictwem magistrali usług korporacyjnych

Magistrala usług korporacyjnych (ESB) umożliwia uzyskanie maksymalnej elastyczności w architekturze SOA. Uczestnicy interakcji usług łączą się z magistralą ESB zamiast bezpośrednio ze sobą.

Kiedy requester usług nawiązuje połączenie z magistralą ESB, magistrala ESB przejmuje odpowiedzialność za dostarczanie jego żądań do dostawcy usług za pomocą komunikatów, realizując w ten sposób wymaganą funkcję i zapewniając jakość usług. Magistrala ESB ułatwia interakcje między requesterami i dostawcami oraz rozwiązuje problemy związane z niezgodnymi protokołami, wzorcami interakcji lub możliwościami usług. Magistrala ESB może również umożliwiać lub usprawniać monitorowanie i zarządzanie. Magistrala ESB udostępnia opcje wirtualizacji i zarządzania, które implementują i rozszerzają podstawowe możliwości architektury SOA.

Magistrala ESB zapewnia obsługę następujących kwestii:

Położenie i tożsamość

Uczestnicy nie muszą znać położenia ani tożsamości innych uczestników. Requestery nie muszą na przykład wiedzieć, że żądanie może być obsłużone przez dowolnego spośród kilku dostawców, a dostawcy usług mogą być dodawani lub usuwani bez przerw w pracy.

Protokół interakcji

Uczestnicy nie muszą współużytkować tego samego protokołu komunikacyjnego ani stylu interakcji. Na

przykład żądanie wyrażane w postaci komunikatu SOAP przez protokół HTTP może być obsługane przez dostawcę, który umożliwia stosowanie tylko protokołu SOAP za pośrednictwem usługi JMS (Java Message Service).

Interfejs

Requestery i dostawcy nie muszą korzystać ze wspólnego interfejsu. Magistrala ESB eliminuje różnice, transformując komunikaty żądań i odpowiedzi na format oczekiwany przez dostawcę.

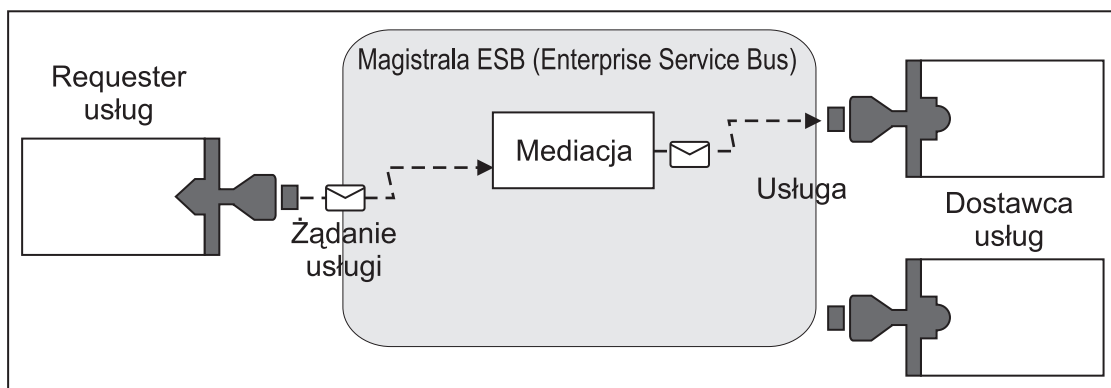
Jakość usług (interakcji)

Uczestnicy lub administratorzy systemu deklarują swoje oczekiwania w zakresie jakości usług, między innymi autoryzację żądań, szyfrowanie i deszyfrowanie treści komunikatów, automatyczną kontrolę interakcji usług oraz sposób kierowania żądań (na przykład optymalizacja pod kątem szybkości lub kosztu).

Umieszczenie magistrali ESB między uczestnikami pozwala modulować ich interakcję za pośrednictwem konstrukcji logicznej zwanej *mediacją*. Mediacje przeprowadzają operacje na wiadomościach w trakcie przetwarzania, kiedy są one przesyłane między requesterami i dostawcami. Mediacji można na przykład użyć do znalezienia usług o określonych parametrach zgłoszonych przez requester, a także do rozstrzygania różnic interfejsu między requesterami i dostawcami. W przypadku złożonych interakcji mediacje można łączyć sekwencyjnie w łańcuchy.

Przy użyciu mediacji magistrala usług korporacyjnych realizuje następujące działania między requesterem i usługą:

- *Kierowanie* komunikatów między usługami. Magistrala usług korporacyjnych oferuje wspólną infrastrukturę komunikacji, której można użyć do łączenia usług, a więc również funkcji biznesowych, jakie reprezentują. Nie wymaga to od programistów pisania i utrzymywania złożonej logiki połączeń.
- *Przekształcanie* protokołów transportu między requesterem i usługą. Magistrala usług korporacyjnych udostępnia spójny, oparty na standardach sposób integrowania funkcji biznesowych korzystających z różnych standardów informatycznych. Umożliwia to integrację funkcji biznesowych, które w innym przypadku nie mogłyby komunikować się ze sobą, na przykład łączenie aplikacji w silosach znajdujących się w działach lub uczestniczenie aplikacji z różnych firm w interakcji usług.
- *Transformowanie* formatów komunikatów między requesterem i usługą. Magistrala usług korporacyjnych umożliwia funkcjom biznesowym wymianę informacji w różnych formatach. Magistrala gwarantuje, że informacje dostarczane do funkcji biznesowej są w formacie wymaganym przez daną aplikację.
- *Obsługa* zdarzeń biznesowych z odmiennych źródeł. Oprócz wymiany komunikatów w celu obsługi żądań usług magistrala usług korporacyjnych obsługuje również interakcje oparte na zdarzeniach.



Rysunek 87. Magistrala usług korporacyjnych. Magistrala usług korporacyjnych kieruje komunikaty między aplikacjami, które są requesterami lub dostawcami usług. Magistrala przekształca protokoły transportowe i transformuje formaty komunikatów między requesterami i dostawcami. Na tym rysunku każda aplikacja używa innego protokołu (protokoły są reprezentowane przez różne kształty geometryczne konektorów) i używa różnych formatów komunikatów.

Jeśli używana jest magistrala usług korporacyjnych, użytkownik może skupić się na podstawowej działalności zamiast na systemach komputerowych. W razie potrzeby istnieje możliwość zmiany lub dodawania usług na przykład w reakcji na zmiany w wymaganiach biznesowych. Można również dodawać moc obliczeniową do obsługi usług lub nowe

możliwości. Wymaganych zmian można dokonać, rekonfigurując magistralę bez wpływu na istniejące usługi i aplikacje wykorzystujące magistralę lub z niewielkim wpływem na nie.

Infrastruktura przesyłania komunikatów magistrali usług korporacyjnych

Produkt IBM Business Process Manager zawiera możliwości magistrali usług korporacyjnych. Produkt IBM Business Process Manager obsługuje integrację technologii, które są zorientowane na usługi oraz obsługę komunikatów, a także sterowane zdarzeniami. Taka integracja pozwala udostępniać opartą na standardach infrastrukturę przesyłania komunikatów w zintegrowanej magistrali usług korporacyjnych.

Możliwości usług korporacyjnych, które mogą być wykorzystane na potrzeby aplikacji korporacyjnych, udostępniają nie tylko warstwę transportową, ale również obsługę mediacji, która ułatwia interakcję między usługami. Magistrala usług korporacyjnych jest zbudowana na podstawie otwartych standardów oraz architektury zorientowanej na usługi (SOA). Jest ona oparta na stabilnej infrastrukturze Java EE oraz powiązanych usługach platformy udostępnianych przez wdrożenie sieciowe serwera IBM WebSphere Application Server.

Produkt IBM Business Process Manager jest oparty na tej samej technologii, która jest dostępna w produkcie IBM WebSphere Enterprise Service Bus. Ta możliwość stanowi część bazowej funkcjonalności produktu IBM Business Process Manager i do wykorzystania możliwości nie jest wymagana żadna dodatkowa licencja na produkt WebSphere Enterprise Service Bus.

Istnieje jednak możliwość wdrożenia w przedsiębiorstwie dodatkowych autonomicznych licencji produktu WebSphere Enterprise Service Bus, co pozwala rozszerzyć zasięg połączeń rozwiązań integracji procesów opartych na produkcie IBM Business Process Manager. Produkt WebSphere Enterprise Service Bus może być na przykład zainstalowany w miejscu bliższym aplikacji SAP, w celu udostępniania adaptera IBM WebSphere Adapter for SAP oraz w celu transformowania komunikatów SAP zanim informacje zostaną wysłane przez sieć do procesu biznesowego opartego na produkcie IBM Business Process Manager.

Hosty przesyłania komunikatów lub miejsca docelowego kolejki

Host przesyłania komunikatów lub miejsca docelowego kolejki udostępnia funkcję przesyłania komunikatów na serwerze. Serwer staje się hostem miejsca docelowego przesyłania komunikatów, gdy zostanie skonfigurowany jako miejsce docelowe przesyłania komunikatów.

Mechanizm przesyłania komunikatów działa na serwerze. Mechanizm ten udostępnia funkcje przesyłania komunikatów oraz punkt pozwalający aplikacjom na nawiązywanie połączenia z magistralą. Kolejki komunikatów w mechanizmie przesyłania komunikatów są używane przez komunikację asynchroniczną architektury SCA (Service Component Architecture), importy oraz eksporty JMS, a także podczas asynchronicznego przetwarzania wewnętrznego.

Środowisko wdrażania tworzy podczas wdrażania modułów aplikacji połączenie między źródłem komunikatu i celem komunikatu przy użyciu magistrali. Wiedza dotycząca źródła komunikatu oraz celu komunikatu ułatwia określenie typu wymaganego środowiska wdrażania.

Aplikacje mogą przechowywać dane trwale w składnicy danych, która jest zestawem tabel w bazie danych lub schemacie, albo w składnicy plików. Mechanizm przesyłania komunikatów używa instancji źródła danych JDBC do interakcji z tą bazą danych.

Host docelowy przesyłania komunikatów należy skonfigurować podczas definiowania środowiska wdrażania przy użyciu opcji **Serwer** z poziomu Konsoli administracyjnej. Ewentualnie jako host docelowy należy wyznaczyć serwer podczas instalacji oprogramowania.

Składnice danych:

Każdy mechanizm przesyłania komunikatów może wykorzystywać składnicę danych będącą zbiorem tabel w bazie danych lub schematem, w którym zapisywane są dane trwałe.

Wszystkie tabele w składnicy danych są przechowywane w tym samym schemacie bazy danych. Istnieje możliwość utworzenia każdej składnicy danych w oddzielnej bazie danych. Można również utworzyć wiele składnic danych w tej samej bazie danych, w której każda składnica danych wykorzystuje inny schemat.

Mechanizm przesyłania komunikatów korzysta z instancji źródła danych JDBC w celu współdziałania z bazą danych zawierającą składnicę danych dla tego mechanizmu przesyłania komunikatów.

Dostawcy JDBC

Dostawcy JDBC umożliwiają współdziałanie z aplikacjami korzystającymi z relacyjnych baz danych.

Aplikacje używają dostawców JDBC do pracy z relacyjnymi bazami danych. Dostawca JDBC udostępnia określoną klasę implementacji sterownika JDBC w celu uzyskania dostępu do określonego typu bazy danych. Aby utworzyć pulę połączeń do tej bazy danych, należy powiązać źródło danych z dostawcą JDBC. Dostawca JDBC i obiekty źródła danych pod względem funkcjonalnym odpowiadają fabryce połączeń JCA (Java EE Connector Architecture), która udostępnia połączenie z nierelacyjną bazą danych.

Należy zapoznać się z przykładami zarówno typowej konfiguracji środowiska autonomicznego, jak i typowej konfiguracji środowiska wdrażania przedstawionymi w poprzednim temacie.

Więcej informacji o dostawcach JDBC można znaleźć w temacie “Dostawcy JDBC” w Centrum informacyjnym produktu WebSphere Application Server.

Magistrala integracji usług dla produktu IBM Business Process Manager

Magistrala integracji usług to zarządzany mechanizm komunikacji obsługujący integrację usług przy użyciu synchronicznego i asynchronicznego przesyłania komunikatów. W skład magistrali wchodzi połączone mechanizmy przesyłania komunikatów służące do zarządzania zasobami magistrali. Jest to jedna z technologii serwera WebSphere Application Server, na których oparto produkt IBM Business Process Manager.

Pojedyncza magistrala integracji usług i pojedynczy mechanizm przesyłania komunikatów domyślnie korzystają z tego samego schematu bazy danych jako bazy danych produktu. Każde środowisko wdrażania ma swoją własną magistralę. Pojedyncza magistrala ma nazwę **BPM.nazwa_środowiska_wdrażania.Bus**.

Miejsce docelowe magistrali to adres logiczny, do którego może zostać przyłączona aplikacja jako producent, konsument lub zarówno jako producent, jak i konsument. Miejsce docelowe kolejki to miejsce docelowe magistrali używane na potrzeby przesyłania komunikatów w trybie punkt z punktem.

Każda magistrala może zawierać jeden lub kilka elementów, z których każdy to serwer lub klaster.

Topologia magistrali jest fizycznym układem serwerów aplikacji, mechanizmów przesyłania komunikatów oraz menedżerów kolejek produktu WebSphere MQ oraz wzorcem połączeń magistrali i powiązań między nimi. Wspólnie tworzą one magistralę usług korporacyjnych.

Aplikacje usług i moduły usług

Moduł usług jest modułem SCA (Service Component Architecture), który udostępnia usługi w środowisku wykonawczym. Po wdrożeniu modułu usług w produkcie IBM Business Process Manager budowana jest powiązana aplikacja usług, która jest pakowana jako plik archiwum korporacyjnego (EAR).

Moduły usług są podstawowymi jednostkami wdrażania i mogą zawierać komponenty, biblioteki oraz moduły pomostowe używane przez powiązaną aplikację usług. Moduły usług mają eksporty oraz opcjonalnie importy, które definiują relacje między modułami oraz requesterami i dostawcami usług. Komponent WebSphere Process Server obsługuje moduły dla usług biznesowych i modułów mediacji. Zarówno moduły jak i moduły mediacji są typami modułów SCA. Moduł mediacji umożliwia komunikację między aplikacjami, wykonując transformację wywołania usługi do formatu zrozumiałego w miejscu docelowym, przekazując żądanie do miejsca docelowego i zwracając wynik do nadawcy. Moduł usługi biznesowej implementuje logikę procesu biznesowego. Jednak moduł może zawierać tę samą logikę mediacji, którą można umieścić w module mediacji.

Wdrażanie aplikacji usług

Proces wdrażania pliku EAR zawierającego aplikację usług jest taki sam jak proces wdrażania dowolnego pliku EAR. Istnieje możliwość modyfikowania wartości dla parametrów mediacji w czasie wdrażania. Po wdrożeniu pliku EAR zawierającego moduł SCA można wyświetlić szczegóły aplikacji usług oraz powiązanego z nią modułu. Użytkownik może zobaczyć, w jaki sposób moduł usług jest połączony z requesterami usług (za pośrednictwem eksportów) oraz dostawcami usług (za pośrednictwem importów).

Wyświetlanie szczegółów modułu SCA

Dostępne do wyświetlenia szczegóły modułu usług są zależne od modułu SCA. Zawierają one następujące atrybuty:

- nazwa modułu SCA,
- opis modułu SCA,
- nazwę powiązanej aplikacji,
- informacje o wersji modułu SCA, jeśli moduł podlega kontroli wersji,
- importy modułu SCA:
 - Interfejsy importu są abstrakcyjnymi definicjami opisującymi sposób uzyskiwania dostępu do usługi przez moduł SCA.
 - Powiązania importu są konkretnymi definicjami określającymi mechanizm fizyczny używany przez moduł SCA w celu uzyskania dostępu do usługi. Przykład: SOAP/HTTP.
- eksporty modułu SCA:
 - Interfejsy eksportu są abstrakcyjnymi definicjami, które opisują sposób uzyskiwania dostępu do modułu SCA przez requestery usług.
 - Powiązania eksportu są konkretnymi definicjami określającymi mechanizm fizyczny używany przez requester usług w celu uzyskania dostępu do modułu SCA oraz pośrednio do usługi.
- właściwości modułu SCA.

Importy i powiązania importu

Importy definiują interakcje między modułami SCA a dostawcami usług. Moduły SCA wykorzystują importy, aby umożliwić komponentom uzyskiwanie dostępu do usług zewnętrznych (usług znajdujących się poza modułem SCA) przy użyciu reprezentacji lokalnej. Powiązania importu definiują konkretny sposób, w jaki uzyskiwany jest dostęp do usługi zewnętrznej.

Jeśli moduły SCA nie wymagają dostępu do usług zewnętrznych, nie muszą mieć importów. Moduły mediacji zazwyczaj mają co najmniej jeden import, który służy do przekazywania komunikatów lub żądań do miejsc docelowych.

Interfejsy i powiązania

Import modułu SCA wymaga co najmniej jednego interfejsu i ma jedno powiązanie.

- Interfejsy importu są abstrakcyjnymi definicjami określającymi zbiór operacji przy użyciu języka WSDL (Web Services Description Language). Jest to język XML służący do opisywania usług Web Services. Moduł SCA może mieć wiele interfejsów importu.
- Powiązania importu są konkretnymi definicjami określającymi mechanizm fizyczny używany przez moduły SCA w celu uzyskania dostępu do usługi zewnętrznej.

Obsługiwane powiązania importu

Produkt IBM Business Process Manager obsługuje następujące powiązania importu:

- Powiązania SCA łączą moduły SCA z innymi modułami SCA. Powiązania SCA są zwane również powiązaniami domyślnymi.

- Powiązania usług Web Services umożliwiają komponentom wywoływanie usług Web Services. Obsługiwane są protokoły SOAP1.1/HTTP, SOAP1.2/HTTP i SOAP1.1/JMS.
Istnieje możliwość użycia powiązania SOAP1.1/HTTP lub SOAP1.2/HTTP na podstawie interfejsu JAX-WS (Java API for XML Web Services). Umożliwia to interakcję z usługami wykorzystującymi powiązania dokumentu lub powiązania literału RPC i wykorzystuje procedury obsługi JAX-WS do dostosowywania wywołań. Dostępne jest oddzielne powiązanie SOAP1.1/HTTP, które umożliwia interakcję z usługami używającymi powiązań kodowanych RPC, oraz które może być stosowane w sytuacji, kiedy wymagane jest używanie procedur obsługi JAX-RPC do dostosowywania wywołań.
- Powiązania HTTP umożliwiają dostęp do aplikacji za pośrednictwem protokołu HTTP.
- Powiązania importu EJB (Enterprise JavaBeans) umożliwiają komponentom SCA wywoływanie usług udostępnianych przez logikę biznesową Java EE działającą na serwerze Java EE.
- Powiązania EIS (Enterprise Information System) udostępniają połączenia między komponentami SCA i zewnętrznym systemem EIS. Ta komunikacja jest realizowana za pośrednictwem adapterów zasobów.
- Powiązania JMS (Java Message Service) 1.1 umożliwiają współdziałanie z domyślnym dostawcą przesyłania komunikatów serwera WebSphere Application Server. Powiązania JMS mogą wykorzystywać różne typy transportu, między innymi TCP/IP, HTTP lub HTTPS. Obsługiwana jest automatycznie klasa komunikatu JMS oraz jej pięć podtypów (tekst, bajty, obiekt, strumień i odwzorowanie).
- Ogólne powiązania JMS umożliwiają współdziałanie z dostawcami JMS innych firm, które są zintegrowane z serwerem WebSphere Application Server przy użyciu narzędzia serwera aplikacji JMS (ASF).
- Powiązania WebSphere MQ JMS umożliwiają współdziałanie z dostawcami JMS opartymi na produkcie WebSphere MQ. Obsługiwana jest automatycznie klasa komunikatu JMS oraz jej pięć podtypów (tekst, bajty, obiekt, strumień i odwzorowanie). Aby używać produktu WebSphere MQ jako dostawcy JMS, należy skorzystać z powiązań JMS WebSphere MQ.
- Powiązania WebSphere MQ umożliwiają współdziałanie z produktem WebSphere MQ. Powiązań WebSphere MQ można używać tylko ze zdalnymi menedżerami kolejek za pośrednictwem połączenia klienta produktu WebSphere MQ. Nie można używać ich z lokalnymi menedżerami kolejek. Aby komunikować się z rodzimymi aplikacjami WebSphere MQ, należy użyć powiązań WebSphere MQ.

Dynamiczne wywoływanie usług

Usługi można wywoływać za pośrednictwem dowolnego obsługiwane powiązania importu. Usługa zazwyczaj znajduje się w punkcie końcowym określonym w imporcie. Ten punkt końcowy jest nazywany statycznym punktem końcowym. Istnieje możliwość wywołania innej usługi, poprzez nadpisanie statycznego punktu końcowego. Dynamiczne nadpisywanie statycznych punktów końcowych pozwala wywołać usługę w innym punkcie końcowym za pośrednictwem dowolnego obsługiwane powiązania importu. Dynamiczne wywoływanie usług pozwala również wywołać usługę w sytuacji, gdy obsługiwane powiązanie importu nie ma statycznego punktu końcowego.

Do określenia protokołu i jego konfiguracji na potrzeby wywołania dynamicznego używany jest import z powiązaniem. Import używany na potrzeby wywołania dynamicznego może być połączony z komponentem wywołującym lub może być wybierany dynamicznie w czasie wykonywania.

W przypadku wywołań usług Web Services i wywołań SCA można również wykonać wywołanie dynamiczne bez importu, na podstawie protokołu i konfiguracji pobranej z adresu URL punktu końcowego. Typ miejsca docelowego wywołania jest identyfikowany na podstawie adresu URL punktu końcowego. Jeśli używany jest import, adres URL musi być zgodny z protokołem powiązania importu.

- Adres URL architektury SCA oznacza wywołanie innego modułu SCA.
- Adres URL protokołu HTTP lub usługi JMS domyślnie oznacza wywołanie usługi Web Service. W przypadku tych adresów URL można podać dodatkową wartość typu powiązania, która określa, czy adres URL reprezentuje wywołanie za pośrednictwem powiązania HTTP czy powiązania JMS.
- W przypadku adresu URL protokołu URL dla usługi Web Service domyślną opcją jest użycie protokołu SOAP 1.1, ale można określić wartość typu powiązania, która wskazuje że należy użyć protokołu SOAP 1.2.

Eksporty i powiązania eksportów

Eksporty definiują interakcje między modułami SCA a requesterami usług. Moduły SCA wykorzystują eksporty w celu udostępniania usług. Powiązania eksportu definiują konkretny sposób, w jaki requestery usług uzyskują dostęp do modułu SCA.

Interfejsy i powiązania

Eksport modułu SCA wymaga co najmniej jednego interfejsu.

- Interfejsy eksportu są abstrakcyjnymi definicjami określającymi zbiór operacji przy użyciu języka WSDL (Web Services Description Language). Jest to język XML służący do opisywania usług Web Services. Moduł SCA może mieć wiele interfejsów eksportu.
- Powiązania eksportu są konkretnymi definicjami określającymi mechanizm fizyczny używany przez requestery usług w celu uzyskania dostępu do usługi. Eksport modułu SCA ma zazwyczaj zdefiniowane jedno powiązanie. Eksport bez określonego powiązania jest interpretowany przez środowisko wykonawcze jako eksport z powiązaniem SCA.

Obsługiwane powiązania eksportu

Produkt IBM Business Process Manager obsługuje następujące powiązania eksportu:

- Powiązania SCA łączą moduły SCA z innymi modułami SCA. Powiązania SCA są zwane również powiązaniami domyślnymi.
- Powiązanie usługi Web Service umożliwia wywoływanie eksportów jako usług Web Services. Obsługiwane są protokoły SOAP1.1/HTTP, SOAP1.2/HTTP i SOAP1.1/JMS.
Istnieje możliwość użycia powiązania SOAP1.1/HTTP lub SOAP1.2/HTTP na podstawie interfejsu JAX-WS (Java API for XML Web Services). Umożliwia to interakcję z usługami wykorzystującymi powiązania dokumentu lub powiązania literału RPC i wykorzystuje procedury obsługi JAX-WS do dostosowywania wywołań. Dostępne jest oddzielne powiązanie SOAP1.1/HTTP, które umożliwia interakcję z usługami używającymi powiązań kodowanych RPC, oraz które może być stosowane w sytuacji, kiedy wymagane jest używanie procedur obsługi JAX-RPC do dostosowywania wywołań.
- Powiązania HTTP umożliwiają dostęp do eksportów za pośrednictwem protokołu HTTP.
- Powiązania EJB (Enterprise JavaBeans) umożliwiają ujawnianie komponentów SCA jako komponentów EJB, co pozwala na wywoływanie komponentów SCA przez logikę biznesową Java EE (komponenty te nie są dostępne dla tej logiki w innej sytuacji).
- Powiązania EIS (Enterprise Information System) udostępniają połączenia między komponentami SCA i zewnętrznym systemem EIS. Ta komunikacja jest realizowana za pośrednictwem adapterów zasobów.
- Powiązania JMS (Java Message Service) 1.1 umożliwiają współdziałanie z domyślnym dostawcą przesyłania komunikatów serwera WebSphere Application Server. Powiązania JMS mogą wykorzystywać różne typy transportu, między innymi TCP/IP, HTTP lub HTTPS. Obsługiwana jest automatycznie klasa komunikatu JMS oraz jej pięć podtypów (tekst, bajty, obiekt, strumień i odwzorowanie).
- Ogólne powiązania JMS umożliwiają współdziałanie z dostawcami JMS innych firm, które są zintegrowane z serwerem WebSphere Application Server przy użyciu narzędzia serwera aplikacji JMS (ASF).
- Powiązania WebSphere MQ JMS umożliwiają współdziałanie z dostawcami JMS opartymi na produkcie WebSphere MQ. Obsługiwana jest automatycznie klasa komunikatu JMS oraz jej pięć podtypów (tekst, bajty, obiekt, strumień i odwzorowanie). Aby używać produktu WebSphere MQ jako dostawcy JMS, należy skorzystać z powiązań JMS WebSphere MQ.
- Powiązania WebSphere MQ umożliwiają współdziałanie z produktem WebSphere MQ. Do nawiązania połączenia z menedżerem kolejek MQ na zdalnym komputerze wykorzystywane jest połączenie zdalne (połączenie klienta). Połączenie lokalne (połączenie powiązania) jest połączeniem bezpośrednim z produktem WebSphere MQ. Może ono być używane tylko do nawiązania połączenia z menedżerem kolejek MQ na tym samym komputerze. Produkt WebSphere MQ pozwala stosować oba typy połączeń, ale powiązania MQ obsługują tylko połączenie zdalne (połączenie klienta).

Moduły mediacji

Moduły mediacji są to moduły SCA (Service Component Architecture), które umożliwiają zmianę formatu, treści oraz miejsca docelowego żądań usług.

Moduły mediacji działają na komunikatach, które są w trakcie przetwarzania między requesterami usług i dostawcami usług. Użytkownik może kierować komunikaty do różnych dostawców usług oraz zmieniać treść lub format komunikatu. Moduły mediacji mogą udostępniać funkcje, takie jak rejestrowanie komunikatów czy przetwarzanie błędów, dostosowane do konkretnych wymagań.

Niektóre aspekty modułów mediacji można zmieniać z Konsoli administracyjnej bez konieczności ponownego wdrażania modułu.

Komponenty modułów mediacji

Moduły mediacji zawierają następujące elementy:

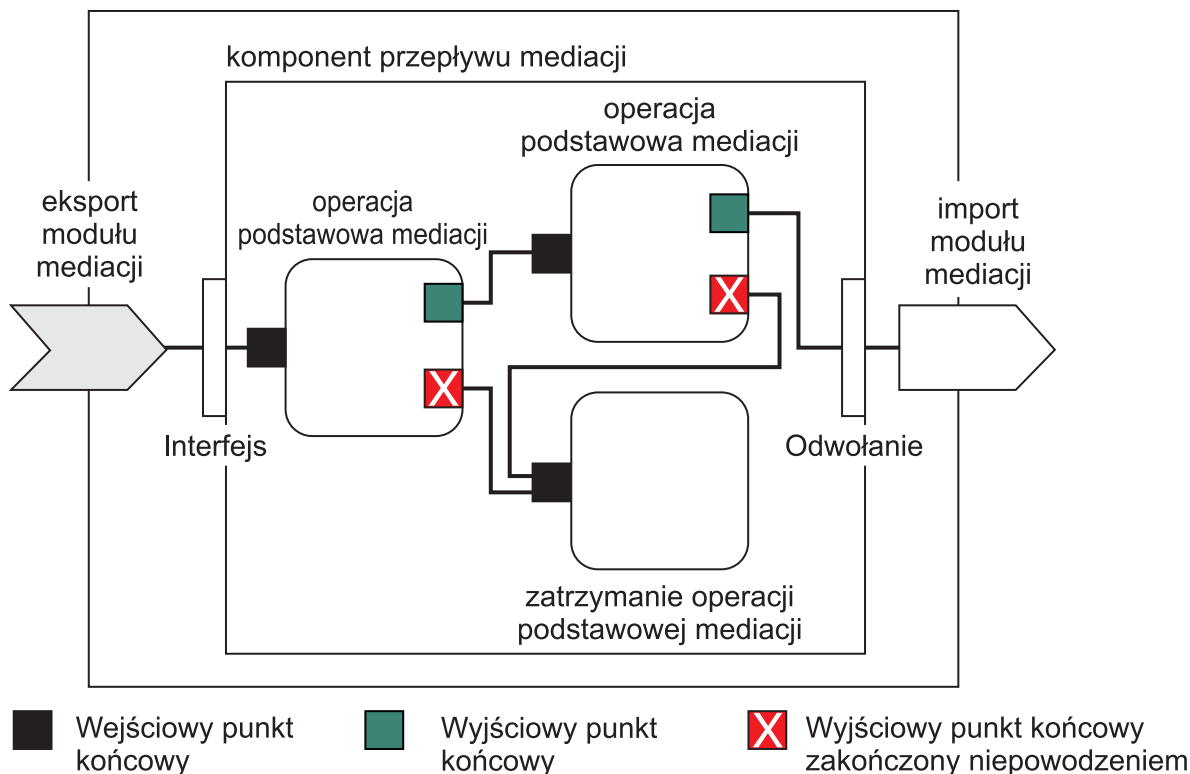
- Importy, które definiują transakcje między modułami SCA i dostawcami usług. Umożliwiają one modułom SCA wywoływanie usług zewnętrznych w taki sposób, jakby były one usługami lokalnymi. Użytkownik może wyświetlać importy modułów mediacji oraz modyfikować powiązanie.
- Eksporty, które definiują interakcje między modułami SCA a requesterami usług. Umożliwiają one modułom SCA oferowanie usług i definiowanie interfejsów zewnętrznych (punktów dostępu) modułu SCA. Użytkownik może wyświetlać eksporty modułu mediacji.
- Komponenty SCA, które stanowią elementy składowe modułów SCA, na przykład modułów mediacji. Istnieje możliwość graficznego tworzenia i dostosowywania modułów SCA przy użyciu produktu Integration Designer. Po wdrożeniu modułu mediacji można dostosować niektóre aspekty tego modułu z poziomu Konsoli administracyjnej bez konieczności ponownego wdrażania modułu.

Moduły mediacji zazwyczaj zawierają konkretny typ komponentu SCA nazywany *komponentem przepływu mediacji*. Komponenty przepływu mediacji definiują przepływy mediacji.

Komponent przepływu mediacji może zawierać dowolną liczbę operacji podstawowych mediacji (żadnych operacji, jedną operację lub określoną liczbę operacji). IBM Business Process Manager obsługuje dostarczany zestaw operacji podstawowych mediacji, które udostępniają funkcje do kierowania i transformowania komunikatów. Aby zwiększyć elastyczność w zakresie operacji podstawowej mediacji, można użyć niestandardowej operacji podstawowej mediacji w celu wywołania niestandardowej logiki.

Celem modułu mediacji, który nie zawiera komponentu przepływu mediacji, jest transformowanie żądań usług z jednego protokołu na inny. Można na przykład utworzyć żądanie usługi przy użyciu protokołu SOAP/JMS i przed przesłaniem dalej przekształcić je do formatu SOAP/HTTP.

Uwaga: Z poziomu produktu IBM Business Process Manager można wyświetlać moduły mediacji i dokonywać w nich pewnych zmian. Z poziomu produktu IBM Business Process Manager nie można jednak wyświetlać ani zmieniać komponentów SCA znajdujących się w module. Do dostosowywania komponentów SCA służy produkt Integration Designer.



Rysunek 88. Uproszczony przykład modułu mediacji. Moduł mediacji zawiera jeden komponent przepływu mediacji, który zawiera operacje podstawowe mediacji.

- Właściwości

Operacje podstawowe mediacji mają właściwości. Niektóre z tych właściwości mogą być wyświetlane w Konsoli administracyjnej jako dodatkowe właściwości modułu SCA.

Aby właściwości operacji podstawowej mediacji były widoczne z poziomu Konsoli administracyjnej produktu IBM Business Process Manager, programista integracji musi awansować właściwości. Niektóre właściwości umożliwiają konfigurowanie przez administratora, a produkt Integration Designer opisuje te właściwości jako właściwości możliwe do awansowania, ponieważ mogą być one awansowane z cyklu integracji do cyklu administracyjnego. Inne właściwości nie nadają się do konfigurowania przez administratora, ponieważ ich modyfikowanie może wpłynąć na przepływ mediacji w taki sposób, że będzie konieczne ponowne wdrożenie modułu mediacji. Program Integration Designer wyświetla właściwości, które można zmienić, aby wypromować właściwości operacji podstawowej mediacji.

Istnieje możliwość użycia Konsoli administracyjnej produktu IBM Business Process Manager do zmiany wartości awansowanych właściwości bez konieczności ponownego wdrażania modułu mediacji ani restartowania serwera lub modułu.

Najczęściej przepływy mediacji korzystają natychmiast ze zmian właściwości. Jeśli jednak zmiany właściwości występują w komórce menedżera wdrażania, są one uwzględniane w każdym węźle po synchronizacji tego węzła. Przepływy mediacji, które są w trakcie przetwarzania, nadal używają wcześniejszych wartości.

Uwaga: Z poziomu Konsoli administracyjnej można zmienić tylko wartości właściwości, ale nie można zmienić grup właściwości, ich nazw ani typów. Aby zmienić grupy właściwości, ich nazwy lub typy, należy użyć produktu Integration Designer.

- Moduł mediacji lub zależna biblioteka mogą również definiować przepływy podrzędne. Przepływ podrzędny hermetyzuje zbiór operacji podstawowych mediacji połączonych ze sobą jako element logiki integracji wielokrotnego użytku. W celu wywołania przepływu podrzędnego do przepływu mediacji można dodać operację podstawową.

Wdrażanie modułów mediacji

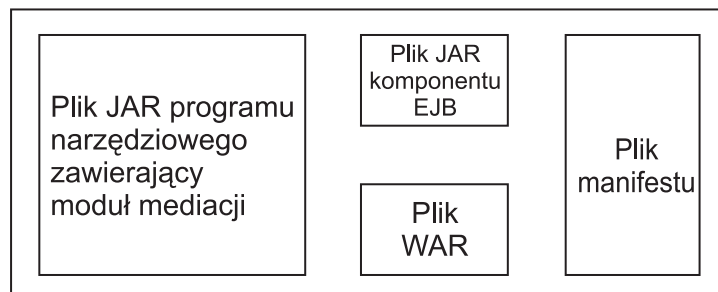
Moduły mediacji są tworzone przy użyciu produktu Integration Designer i są zazwyczaj wdrażane w produkcie IBM Business Process Manager wewnątrz pliku archiwum korporacyjnego (EAR).

Wartość awansowanych właściwości można zmieniać w czasie wdrażania.

Moduł mediacji można wyeksportować z produktu Integration Designer, co powoduje, że produkt Integration Designer spakuje moduł mediacji wewnątrz pliku archiwum Java (JAR), a plik JAR wewnątrz pliku EAR. Plik EAR można następnie wdrożyć, instalując nową aplikację z poziomu Konsoli administracyjnej.

Moduły mediacji mogą być postrzegane jako pojedynczy obiekt. Moduły SCA są jednak definiowane przez kilka plików XML zapisanych w pliku JAR.

Przykład pliku EAR zawierającego moduł mediacji



Rysunek 89. Uproszczony przykład pliku EAR zawierającego moduł mediacji. Plik EAR zawiera pliki JAR. Plik programu narzędziowego JAR zawiera moduł mediacji.

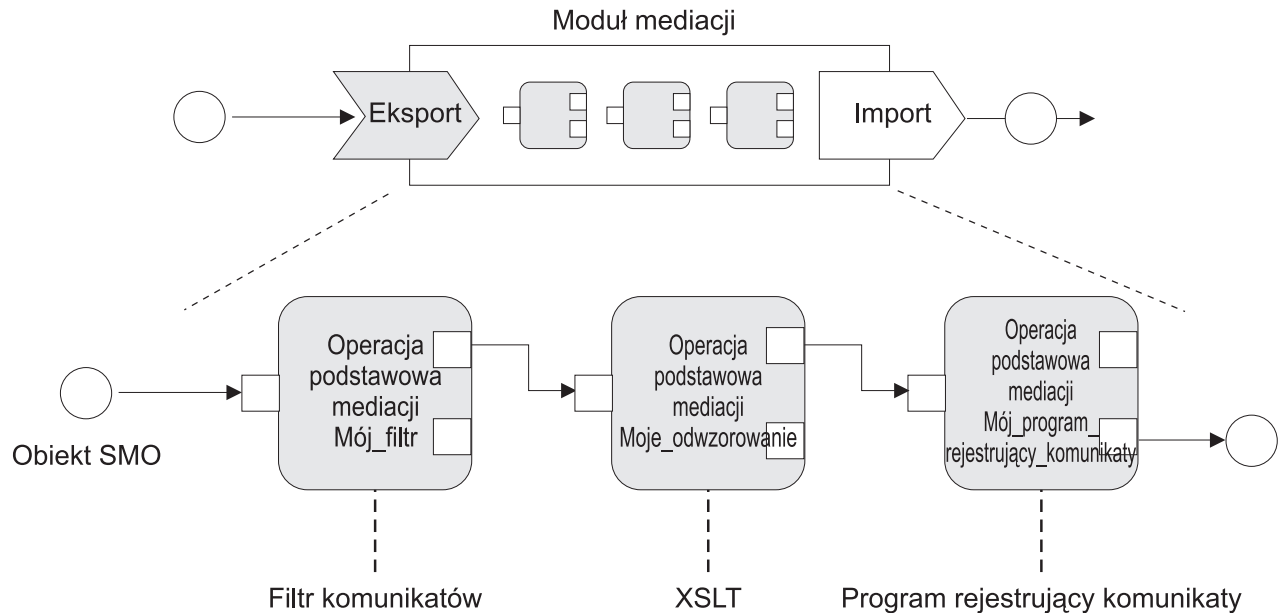
Operacje podstawowe mediacji

Komponenty przepływu mediacji opierają się na przepływach komunikatów między komponentami usług. Możliwości komponentu mediacji są implementowane przez *operacje podstawowe mediacji*, które implementują standardowe typy implementacji usług.

Komponent przepływu mediacji ma co najmniej jeden przepływ. Na przykład: jeden przepływ dla żądania i jeden dla odpowiedzi.

IBM Business Process Manager obsługuje dostarczany zbiór operacji podstawowych mediacji, które implementują standardowe możliwości mediacji dla modułów mediacji lub modułów wdrożonych w produkcie IBM Business Process Manager. Jeśli potrzebne są specjalne możliwości mediacji, użytkownik może utworzyć własne niestandardowe operacje podstawowe mediacji.

Operacja podstawowa mediacji definiuje operację “wejścia”, która przetwarza lub obsługuje komunikaty reprezentowane przez obiekty komunikatów usługi (Service Message Object - SMO). Operacja podstawowa mediacji może również definiować operacje “wyjścia”, które wysyłają komunikaty do innego komponentu lub modułu.



Rysunek 90. Moduł mediacji zawierający trzy operacje podstawowe mediacji

Produktu Integration Designer można użyć do konfigurowania operacji podstawowych mediacji i ustawiania ich właściwości. Niektóre z tych właściwości mogą być uwidaczniane administratorowi środowiska wykonawczego poprzez ich awansowanie. Dowolna właściwość operacji podstawowej mediacji, która może być awansowana, może być również właściwością dynamiczną. Właściwość dynamiczna może być nadpisana w czasie wykonywania przy użyciu pliku strategii.

Integration Designer umożliwia również graficzne modelowanie i składanie komponentów przepływu mediacji z operacji podstawowych mediacji oraz składanie modułów mediacji lub modułów na podstawie komponentów przepływu mediacji. Konsola administracyjna odnosi się do modułów mediacji i modułów jako modułów SCA.

Integration Designer umożliwia również definiowanie przepływów podrzędnych w modułach lub zależnych od nich bibliotekach. Przepływ podrzędny może zawierać dowolną operację podstawową mediacji oprócz operacji podstawowej mediacji Rozstrzygnięcie strategii. Przepływ podrzędny jest wywoływany z poziomu przepływu żądania lub odpowiedzi lub z poziomu innego przepływu podrzędnego przy użyciu operacji podstawowej mediacji Przepływ podrzędny. Właściwości awansowane z operacji podstawowych mediacji w przepływie podrzędnym są ujawniane jako właściwości w operacjach podstawowych mediacji Przepływ podrzędny. Mogą być następnie ponownie awansowane, aż do poziomu modułu, gdzie mogą być modyfikowane przez administratora środowiska wykonawczego.

Obsługiwane operacje podstawowe mediacji

IBM Business Process Manager obsługuje następujący zbiór operacji podstawowych mediacji:

Odzworowanie obiektu biznesowego

Transformuje komunikaty.

- Definiuje transformacje komunikatów przy użyciu odwzorowania obiektu biznesowego, które może być ponownie wykorzystywane.
- Umożliwia graficzne definiowanie transformacji komunikatów przy użyciu edytora odwzorowań obiektów biznesowych.
- Może zmieniać treść komunikatu.
- Może transformować typ komunikatu wejściowego na inny typ komunikatu wyjściowego.

Mediacja niestandardowa

Umożliwia zaimplementowanie własnej logiki mediacji w kodzie Java. Operacja podstawowa mediacji

Mediacja niestandardowa łączy elastyczność operacji podstawowej mediacji zdefiniowanej przez użytkownika z prostotą predefiniowanej operacji podstawowej mediacji. Istnieje możliwość tworzenia złożonych transformacji i wzorców kierowania poprzez:

- Tworzenie kodu Java.
- Tworzenie własnych właściwości.
- Dodawanie nowych punktów końcowych.

Z poziomu operacji podstawowej mediacji Mediacja niestandardowa można wywoływać usługę, ale do wywoływania usług zaprojektowana jest operacja podstawowa mediacji Wywołanie usługi, która udostępnia dodatkowe funkcje takie jak ponawianie prób.

Procedura obsługi danych

Umożliwia transformowanie części komunikatu. Służy do przekształcenia elementu komunikatu z formatu fizycznego w strukturę logiczną lub odwrotnie. Podstawowym zastosowaniem tej operacji podstawowej mediacji jest przekształcanie formatu fizycznego, na przykład łańcucha Tekst w obiekcie komunikatu tekstu JMS, w logiczną strukturę obiektu biznesowego i odwrotnie. Ta mediacja jest często używana do realizowania następujących działań:

- Transformowania sekcji komunikatu wejściowego ze zdefiniowanej struktury na inną. Przykład: jeśli obiekt SMO zawiera wartość łańcucha rozdzielaną przecinkami, a użytkownik chce przeanalizować tę wartość i umieścić dane w konkretnym obiekcie biznesowym.
- Zmiany typu komunikatu. Przykład: jeśli eksport JMS został skonfigurowany tak, aby używał powiązania danych podstawowego typu JMS, a programista integracji zdecyduje, że treść wewnątrz modułu mediacji powinna zostać przekształcona w konkretną strukturę obiektu biznesowego.

Wyszukiwanie w bazie danych

Modyfikuje komunikaty przy użyciu informacji z bazy danych dostarczonej przez użytkownika.

- Konieczne jest skonfigurowanie bazy danych, źródła danych oraz wszystkich ustawień uwierzytelniania na serwerze, które mają być używane przez operację podstawową mediacji Wyszukiwanie w bazie danych. W tym celu należy użyć Konsoli administracyjnej.
- Operacja podstawowa mediacji Wyszukiwanie w bazie danych może odczytywać dane z tylko jednej tabeli.
- Określona kolumna klucza musi zawierać wartości unikalne.
- Dane w kolumnach wartości muszą być danymi typu prostego schematu XML lub typu schematu XML, który rozszerza typ prostego schematu XML.

Wyszukiwanie punktów końcowych

Umożliwia dynamiczne kierowanie żądań poprzez wyszukiwanie punktów końcowych usług w repozytorium.

- Informacje o punktach końcowych usług są pobierane z produktu WebSphere Service Registry and Repository (WSRR). Rejestr WSRR może być rejestrem lokalnym lub zdalnym.
- Zmiany w rejestrze są dokonywane z poziomu Konsoli administracyjnej produktu WSRR.
- IBM Business Process Manager musi wiedzieć, jaki rejestr ma być wykorzystywany, a więc należy utworzyć definicje dostępu do rejestru WSRR przy użyciu Konsoli administracyjnej produktu IBM Business Process Manager.

Emiter zdarzeń

Rozszerza monitorowanie poprzez możliwość wysyłania zdarzeń z wewnątrz komponentu przepływu mediacji.

- Działanie mediowania można zawiesić, anulując zaznaczenie pola wyboru.
- Zdarzenia emitera zdarzeń można wyświetlać przy użyciu przeglądarki modelu CBE (Common Base Events) w produkcie IBM Business Process Manager.
- Zdarzenia należy wysyłać jedynie w znaczącym punkcie przepływu mediacji (z powodów związanych z wydajnością).
- Istnieje możliwość zdefiniowania części komunikatu zawartego w zdarzeniu.
- Zdarzenia są wysyłane w postaci zdarzeń modelu CBE (Common Base Events) i są wysyłane do serwera infrastruktury CEI (Common Event Infrastructure).

- Aby w pełni wykorzystać informacje Emitera zdarzeń, konsumenci zdarzeń muszą rozumieć strukturę zdarzeń CBE. Zdarzenia CBE mają ogólny schemat, ale nie jest to model danych konkretnej aplikacji, które są zawarte w elementach danych rozszerzonych. Aby modelować elementy danych rozszerzonych, narzędzia produktu Integration Designer generują plik definicji katalogu zdarzeń infrastruktury CEI (Common Event Infrastructure) dla każdej skonfigurowanej operacji podstawowej mediacji Emiter zdarzeń. Pliki definicji katalogu zdarzeń są artefaktami eksportu, które są udostępniane w celu pomocy użytkownikowi. Nie są one używane przez produkt Integration Designer ani przez środowisko wykonawcze produktu IBM Business Process Manager. Użytkownik powinien odwoływać się do plików definicji katalogu zdarzeń podczas tworzenia aplikacji w celu konsumowania zdarzeń Emitera zdarzeń.
- Istnieje możliwość określenia innego monitorowania z poziomu produktu IBM Business Process Manager. Można na przykład monitorować zdarzenia, które mają być emitowane z importów i eksportów.

Niepowodzenie

Zatrzymuje konkretną ścieżkę z przepływie i generuje wyjątek.

Zwielokrotnianie wejściowe

Pomaga agregować (składać) komunikaty.

- Może być używana tylko w połączeniu z operacją podstawową mediacji Zwielokrotnianie wyjściowe.
- Operacje podstawowe mediacji Zwielokrotnianie wejściowe i Zwielokrotnianie wyjściowe wspólnie umożliwiają agregowanie danych w pojedynczym komunikacie wyjściowym.
- Operacja podstawowa mediacji Zwielokrotnianie wejściowe odbiera wiadomości, aż zostanie osiągnięty punkt decyzyjny, a następnie tworzony jest jeden komunikat wyjściowy.
- Do przechowywania danych agregacji należy użyć kontekstu współużytkowanego.

Zwielokrotnianie wyjściowe

Pomaga dzielić i agregować (składać) komunikaty.

- Operacje podstawowe mediacji Zwielokrotnianie wejściowe i Zwielokrotnianie wyjściowe wspólnie umożliwiają agregowanie danych w pojedynczym komunikacie wyjściowym.
- W trybie iteracji operacja podstawowa mediacji Zwielokrotnianie wyjściowe umożliwia iterację pojedynczego komunikatu wejściowego zawierającego powtarzający się element. Dla każdego wystąpienia powtarzającego się elementu wysyłany jest komunikat.
- Do przechowywania danych agregacji należy użyć kontekstu współużytkowanego.

Procedura ustawiająca nagłówek HTTP

Udostępnia mechanizm do zarządzania nagłówkami w komunikatach HTTP.

- Umożliwia tworzenie, ustawianie, kopiowanie lub usuwanie nagłówków komunikatu HTTP.
- Umożliwia ustawienie wielu działań w celu zmiany wielu nagłówków HTTP.

Odwzorowanie

Transformuje komunikaty.

- Umożliwia wykonywanie transformacji XSL (Extensible Stylesheet Language) lub transformacji odwzorowania obiektu biznesowego.
- Komunikaty są transformowane przy użyciu transformacji XSLT 1.0 lub XSLT 2.0 bądź transformacji odwzorowania obiektu biznesowego. Transformacje XSL działają na serializowanym komunikacie XML, natomiast transformacje odwzorowania obiektu biznesowego działają na obiektach Service Data Object (SDO).

Procedura ustawiająca elementy komunikatów

Udostępnia prosty mechanizm ustawiania treści komunikatów.

- Umożliwia zmianę, dodawanie lub usuwanie elementów komunikatu.
- Nie zmienia typu komunikatu.
- Dane w kolumnach wartości muszą być danymi typu prostego schematu XML lub typu schematu XML, który rozszerza typ prostego schematu XML.

Filtr komunikatów

Kieruje komunikaty do różnych ścieżek, na podstawie treści komunikatu.

- Działanie mediowania można zawiesić, anulując zaznaczenie pola wyboru.

Program rejestrujący komunikaty

Rejestruje komunikaty w relacyjnej bazie danych lub za pośrednictwem niestandardowego programu rejestrującego użytkownika. Komunikaty są zapisywane w formacie XML, a więc dane mogą być później przetwarzane przez aplikacje obsługujące format XML.

- Działanie mediowania można zawiesić, anulując zaznaczenie pola wyboru.
- Schemat relacyjnej bazy danych (struktura tabel) jest zdefiniowany przez IBM.
- Domyślnie operacja podstawowa mediacji Program rejestrujący komunikaty wykorzystuje wspólną bazę danych. Środowisko wykonawcze odwzorowuje źródło danych **jdbc/mediation/messageLog** na wspólną bazę danych.
- Istnieje możliwość ustawienia klas implementacji procedury obsługi, aby dostosować działanie niestandardowego programu rejestrującego. Można opcjonalnie udostępnić klasy implementacji formatera, klasy implementacji filtru lub obie te klasy, aby dostosować działanie niestandardowego programu rejestrującego.

Procedura ustawiająca nagłówek MQ

Udostępnia mechanizm do zarządzania nagłówkami w komunikatach MQ.

- Umożliwia tworzenie, ustawianie, kopiowanie lub usuwanie nagłówków komunikatu MQ.
- Umożliwia ustawienie wielu działań w celu zmiany wielu nagłówków MQ.

Rozstrzygnięcie strategii

Umożliwia dynamiczne konfigurowanie żądań poprzez wyszukiwanie punktów końcowych usług i powiązanych plików strategii w repozytorium.

- Istnieje możliwość użycia pliku strategii do dynamicznego nadpisywania awansowanych właściwości innych operacji podstawowych mediacji.
- Informacje o punktach końcowych usług i informacje o strategiach są pobierane z produktu WebSphere Service Registry and Repository (WSRR). Rejestr WSRR może być rejestrem lokalnym lub zdalnym.
- Zmiany w rejestrze są dokonywane z poziomu Konsoli administracyjnej produktu WSRR.
- IBM Business Process Manager musi wiedzieć, jaki rejestr ma być wykorzystywany, a więc należy utworzyć definicje dostępu do rejestru WSRR przy użyciu Konsoli administracyjnej produktu IBM Business Process Manager.

Wywołanie usługi

Wywołuje usługę z wewnątrz przepływu mediacji, zamiast czekać na koniec przepływu mediacji i używać mechanizmu wywołania.

- Jeśli usługa zwróci błąd, można ponownie wywołać tę samą usługę lub wywołać inną.
- Operacja podstawowa mediacji Wywołanie usługi jest zaawansowaną operacją podstawową mediacji, której można używać samodzielnie na potrzeby prostych wywołań usług lub w połączeniu z innymi operacjami podstawowymi mediacji w celu realizacji złożonych mediacji.

Ustawianie typu komunikatu

Podczas programowania integracji umożliwia traktowanie pól komunikatów o słabych typach tak, jakby miały stały typ. Pole ma słaby typ, jeśli może zawierać więcej niż jeden typ danych. Pole ma stały typ, jeśli jego typ oraz struktura wewnętrzna są znane.

- W czasie wykonywania operacja podstawowa mediacji Ustawianie typu komunikatu umożliwia sprawdzenie, czy treść komunikatu jest zgodna z oczekiwanymi typami danych.

Procedura ustawiająca nagłówek SOAP

Udostępnia mechanizm do zarządzania nagłówkami w komunikatach SOAP.

- Umożliwia tworzenie, ustawianie, kopiowanie lub usuwanie nagłówków komunikatu SOAP.
- Umożliwia ustawienie wielu działań w celu zmiany wielu nagłówków SOAP.

Zatrzymywanie

Zatrzymuje konkretną ścieżkę w przepływie bez generowania wyjątku.

Filtr typów

Umożliwia kierowanie komunikatów do różnych ścieżek przepływu na podstawie ich typu.

Pobieranie z produktu WebSphere eXtreme Scale

Istnieje możliwość pobierania informacji ze środowiska pamięci podręcznej serwera eXtreme Scale.

- Można wyszukać wartości w pamięci podręcznej i zapisać je jako elementy w komunikacie przy użyciu klucza.
- Łącząc operacje podstawowe mediacji Zapisywanie i Pobieranie z produktu eXtreme Scale, można umieścić w pamięci podręcznej odpowiedź z systemu zaplecza. Późniejsze żądania nie będą musiały uzyskiwać dostępu do tego systemu zaplecza.
- Należy utworzyć definicje produktu eXtreme Scale przy użyciu Konsoli administracyjnej produktu WebSphere ESB, aby umożliwić określenie serwera eXtreme Scale, który powinien być używany.

Zapisywanie w produkcie WebSphere eXtreme Scale

Istnieje możliwość zapisania informacji w środowisku pamięci podręcznej serwera eXtreme Scale.

- Informacje można zapisać w pamięci podręcznej produktu eXtreme Scale przy użyciu klucza i obiektu.
- Łącząc operacje podstawowe mediacji Zapisywanie i Pobieranie z produktu eXtreme Scale, można użyć operacji podstawowej mediacji Zapisywanie do zapisania danych w obrębie pamięci podręcznej, a operacji podstawowej mediacji Pobieranie - do pobrania danych zapisanych wcześniej w pamięci podręcznej.
- Należy utworzyć definicje produktu eXtreme Scale przy użyciu Konsoli administracyjnej produktu WebSphere ESB, aby umożliwić określenie serwera eXtreme Scale, który powinien być używany.

Dynamiczne kierowanie

Komunikaty można kierować na różne sposoby przy użyciu punktów końcowych zdefiniowanych w czasie integracji lub punktów końcowych określanych dynamicznie w czasie wykonywania.

Kierowanie dynamiczne obejmuje dwa następujące przypadki:

- Kierowanie komunikatów w sytuacji, w której przepływ jest dynamiczny, ale wszystkie możliwe punkty końcowe są predefiniowane w module SCA (Service Component Architecture).
- Kierowanie komunikatów w sytuacji, w której przepływ jest dynamiczny i wybór punktów końcowych jest również dynamiczny. Punkty końcowe usługi są wybierane ze źródła zewnętrznego w czasie wykonywania.

Dynamiczny wybór punktu końcowego

Środowisko wykonawcze ma możliwość kierowania komunikatów żądań i odpowiedzi do adresu punktu końcowego identyfikowanego przez element nagłówka komunikatu. Ten element nagłówka komunikatu może być aktualizowany przez operacje podstawowe mediacji w przepływie mediacji. Adres punktu końcowego może być aktualizowany przy użyciu informacji z rejestru, bazy danych lub przy użyciu informacji z samego komunikatu. Kierowanie komunikatów odpowiedzi ma zastosowanie tylko, jeśli odpowiedź jest wysłana przez eksport JAX-WS usługi Web Service.

Aby środowisko wykonawcze implementowało kierowanie dynamiczne dla żądania lub odpowiedzi, moduł SCA musi mieć ustawioną właściwość **Używaj dynamicznego punktu końcowego**, jeśli jest ustawiony w nagłówku komunikatu. Programiści integracji mogą ustawić właściwość **Używaj dynamicznego punktu końcowego**, jeśli jest ustawiony w nagłówku komunikatu lub mogą awansować tę właściwość (uwidocznic ją w czasie wykonywania), aby mógł ją ustawić administrator środowiska wykonawczego. Właściwości modułu można wyświetlić w oknie Właściwości modułu. Aby wyświetlić to okno, należy wybrać opcję **Aplikacje > Moduły SCA > Właściwości modułu**. Programista integracji nadaje awansowanym właściwościom nazwy aliasów i te właśnie nazwy są wyświetlane w Konsoli administracyjnej.

Rejestr

Do przechowywania informacji o punktach końcowych usług można użyć produktu IBM WebSphere Service Registry and Repository (WSRR). Następnie można utworzyć moduły SCA do pobierania punktów końcowych z rejestru WSRR.

Podczas tworzenia modułów SCA użytkownik wykorzystuje operację podstawową mediacji Wyszukiwanie punktów końcowych, która umożliwia przepływowi mediacji wysłanie zapytania do rejestru WSRR w celu uzyskania punktu końcowego usługi lub zbioru punktów końcowych usługi. Jeśli moduł SCA pobiera zbiór punktów końcowych, musi użyć innej operacji podstawowej mediacji w celu wybrania preferowanego punktu końcowego.

Sterowanie żądaniami usług przy użyciu strategii mediacji

Strategii mediacji można używać do sterowania przepływami mediacji między requesterami a dostawcami usług.

Przepływami mediacji można sterować przy użyciu strategii mediacji zapisanych w produkcie IBM WebSphere Service Registry and Repository (WSRR). Implementacja zarządzania strategiami usług w produkcie WSRR jest oparta na środowisku strategii usług Web Services (Web Services Policy Framework - WS-Policy).

Aby sterować żądaniami usług przy użyciu strategii mediacji w rejestrze produktu WSRR muszą znajdować się odpowiednie moduły SCA (Service Component Architecture) i dokumenty strategii mediacji.

Sposób przyłączania strategii mediacji do żądania usługi

Podczas tworzenia modułu SCA, który wymaga użycia strategii mediacji, do przepływu mediacji należy dołączyć operację podstawową mediacji Rozstrzygnięcie strategii. Operacja podstawowa mediacji Rozstrzygnięcie strategii uzyskuje informacje o strategii mediacji z rejestru w czasie wykonywania. Dlatego też moduł SCA musi zawierać komponent przepływu mediacji, aby obsługiwać sterowanie żądaniami usług przy użyciu strategii mediacji.

Istnieje możliwość przyłączenia co najmniej jednej strategii mediacji w rejestrze do modułu SCA lub do usługi docelowej używanej przez moduł SCA. Przyłączone strategie mediacji mogą być używane na potrzeby wszystkich komunikatów usług przetwarzanych przez ten moduł SCA (znajdują się w zasięgu). Strategie mediacji mogą zawierać załączniki strategii, które definiują warunki. Warunki strategii mediacji umożliwiają stosowanie różnych strategii mediacji w różnych kontekstach. Strategie mediacji mogą dodatkowo zawierać klasyfikacje, których można użyć do określenia stanu nadzoru.

WebSphere Service Registry and Repository

Produkt WebSphere Service Registry and Repository (WSRR) umożliwia zapisywanie informacji, uzyskiwanie dostępu do informacji oraz zarządzanie informacjami dotyczącymi punktów końcowych usług oraz strategii mediacji. Produktu WSRR można użyć, aby aplikacje usług były bardziej dynamiczne i łatwiejsze do dostosowania do zmiennych warunków biznesowych.

Wprowadzenie

Przepływy mediacji mogą używać produktu WSRR jako mechanizmu dynamicznego wyszukiwania, udostępniającego informacje o punktach końcowych usług lub strategiach mediacji.

Aby skonfigurować dostęp do produktu WSRR, należy utworzyć dokumenty definicji WSRR przy użyciu Konsoli administracyjnej. Można również użyć komend administrowania produktem WSRR z klienta skryptowego wsadmin. Definicje WSRR oraz właściwości ich połączeń są mechanizmem używanym w celu nawiązywania połączenia z instancją rejestru i pobierania punktu końcowego usługi lub strategii mediacji.

Punkty końcowe usług

Produktu WSRR można użyć do zapisywania informacji o używanych usługach, usługach, których użycie jest planowane lub usługach, o których użytkownik chce mieć informacje. Te usługi mogą znajdować się w systemach użytkownika lub w innych systemach. Aplikacja może na przykład użyć produktu WSRR do znalezienia najbardziej odpowiedniej usługi w celu spełnienia potrzeb funkcjonalnych i związanych z wydajnością.

Podczas tworzenia modułu SCA, który wymaga dostępu do punktów końcowych usług z produktu WSRR, do przepływu mediacji należy dołączyć operację podstawową mediacji Wyszukiwanie punktów końcowych. Operacja podstawowa mediacji Wyszukiwanie punktów końcowych uzyskuje punkty końcowe usług z rejestru w czasie wykonywania.

Strategie mediacji

Produktu WSRR można również użyć do zapisywania informacji o strategii mediacji. Strategie mediacji pomagają sterować żądaniami usług, dynamicznie nadpisując właściwości modułu. Jeśli produkt WSRR zawiera strategię mediacji, które są przyłączone do obiektu reprezentującego moduł SCA lub usługę docelową, strategię mediacji mogą nadpisywać właściwości modułu. Aby stosować różne strategię mediacji w różnych kontekstach, można utworzyć warunki strategii mediacji.

Uwaga: Strategie mediacji dotyczą sterowania przepływem mediacji, a nie zabezpieczeń.

Podczas tworzenia modułu SCA, który wymaga użycia strategii mediacji, do przepływu mediacji należy dołączyć operację podstawową mediacji Rozstrzygnięcie strategii. Operacja podstawowa mediacji Rozstrzygnięcie strategii uzyskuje informacje o strategii mediacji z rejestru w czasie wykonywania.

WebSphere eXtreme Scale

WebSphere eXtreme Scale (eXtreme Scale) umożliwia udostępnianie systemu zapisywania w pamięci podręcznej, który można zintegrować z produktem IBM Business Process Manager. Użycie produktu eXtreme Scale z produktem IBM Business Process Manager pozwala skrócić czasy odpowiedzi i zwiększyć niezawodność, a także udostępnić dodatkowe funkcje integracji.

eXtreme Scale działa jak elastyczna i skalowalna siatka danych w pamięci. Ta siatka danych dynamicznie buforuje, partycjonuje i replikuje dane aplikacji oraz logikę biznesową na wielu serwerach, a także pozwala nimi zarządzać. eXtreme Scale ułatwia także uzyskanie takich właściwości usługi, jak integralność transakcyjna, wysoka dostępność i przewidywalne czasy odpowiedzi.

Dostęp do funkcji buforowania produktu eXtreme Scale można uzyskać, umieszczając w obrębie przepływu operacje podstawowe mediacji produktu WebSphere eXtreme Scale. Podczas tworzenia modułu architektury Service Component Architecture (SCA), który wymaga zapisania informacji w pamięci podręcznej produktu eXtreme Scale należy dołączyć do przepływu mediacji operację podstawową mediacji Zapisywanie do produktu WebSphere eXtreme Scale. Aby pobrać informacje z pamięci podręcznej produktu eXtreme Scale, należy dołączyć operację podstawową mediacji Pobieranie z produktu WebSphere eXtreme Scale. Połączenie tych dwóch operacji podstawowych mediacji w przepływie mediacji umożliwia buforowanie odpowiedzi z systemu zaplecza, aby późniejsze żądania mogły ją pobrać z pamięci podręcznej.

Aby skonfigurować dostęp do produktu eXtreme Scale, należy utworzyć definicję produktu WebSphere eXtreme Scale przy użyciu Konsoli administracyjnej. Można również użyć komend administrowania produktem WebSphere eXtreme Scale z klienta skryptowego wsadmin. Definicja produktu eXtreme Scale jest mechanizmem używanym przez operacje podstawowe mediacji Zapisywanie i Pobieranie produktu WebSphere eXtreme Scale do łączenia się z serwerem eXtreme Scale.

Klienci usługi komunikatów

Dostępne są klienci usługi komunikatów dla środowisk C/C++ i .NET, które umożliwiają aplikacjom w językach innych niż Java nawiązywanie połączenia z magistralą usług korporacyjnych.

Message Service Clients for C/C++ and .NET udostępniają funkcję API o nazwie XMS, która ma ten sam zbiór interfejsów co interfejs API usługi JMS (Java Message Service). Klient usługi komunikatów dla środowiska C/C++ zawiera dwie implementacje funkcji API XMS - jedną do używania w aplikacjach w języku C i jedną do używania w aplikacjach w języku C++. Message Service Client for .NET zawiera w pełni zarządzaną implementację funkcji API XMS, która może być używana przez dowolny język zgodny ze środowiskiem .NET.

Klienci usługi komunikatów dla środowiska .NET można uzyskać pod adresem http://www-01.ibm.com/support/docview.wss?rs=0&q1=IA9H&uid=swg24011756&loc=en_US&cs=utf-8&cc=us&lang=en.

Klienci usługi komunikatów dla środowiska C/C++ można uzyskać pod adresem http://www-01.ibm.com/support/docview.wss?rs=0&q1=ia94&uid=swg24007092&loc=en_US&cs=utf-8&cc=us&lang=en.

Istnieje również możliwość zainstalowania i wykorzystywania obsługi klienta Java EE z produktu WebSphere Application Server Network Deployment. Zawiera on klienta usług Web Services, klienta EJB i klienta JMS.

